A LIBRARY OF SOFTWARE FOR THE **IBM PC** & **PCjr**

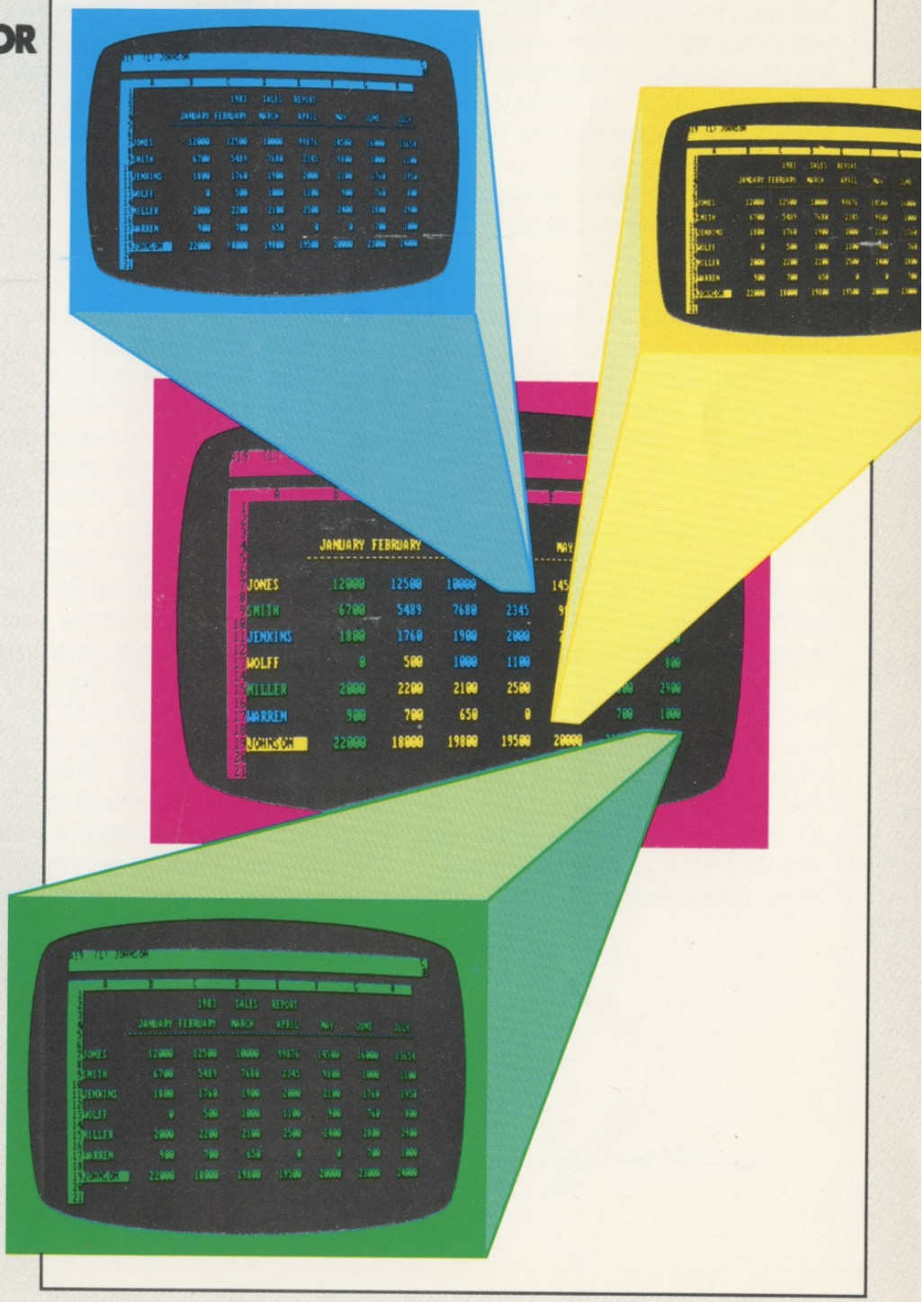
SPECIAL: MICROSOFT CHART DEMO DISK INCLUDED

PC DISK

USER'S MANUAL **VOL. 1 NO. 8** **$34.95**
DISKETTE INCLUDED

SPREADSHEET CONSOLIDATOR

COMBINE VISICALC, LOTUS, AND OTHER SPREADSHEET FILES



PLUS:

STATISTICS LIBRARY

HOME INVENTORY

BASIC SCREENMAKER

STATES AND CAPITALS

BASIC BOOTSTRAPS

SECRET SEQUENCE

DETAILS ON BACKCOVER

FROM THE EDITOR

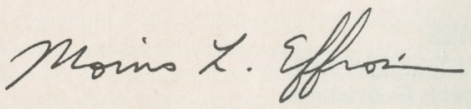
Extra, extra, "read" all about it! Microsoft Corporation, the leading micro-software publisher, includes disk-based demonstration of their new business graphics program CHART, *in latest issue of* PC DISK!

This issue introduces our first advertisement in *PC Disk,* and we are very excited about it. You may have already noticed that this package contains a second diskette, in addition to our regular editorial diskette, along with an accompanying user's manual. These materials represent a new kind of software advertising which gives you the opportunity to try before you buy. Better yet, this opportunity comes at no additional cost. We are pleased indeed that an industry-leader such as Microsoft has recognized *PC Disk* as a medium uniquely suited to bringing this offering to the marketplace. What pleases us most is the additional benefit provided to purchasers of *PC Disk*.

The benefit of such advertising to us is clear, and we make no bones about it. However, we are sincere about the benefit to our readers as well. One of the cruelest dilemmas of microcomputer software today is that you can't try software until you buy it, and once you open a package to try it, you own it. In today's sea of software products, fishing out the right product for your needs can be a very expensive proposition. While it's true that there's a multitude of literature covering software products, experience demonstrates that there is no substitute for a hands-on trial of the software.

The problem is equally troubling to software publishers themselves. Given the fierce competition, they are continually challenged to find new ways to help their product reach its market. An important innovation for buyers and sellers alike is the demo disk, which lets prospective buyers see the software in action. But there is still the problem of finding the time, and a machine, to try the software in a store, or of having to pay for the demo disk itself.

The time for disk-based advertising has clearly arrived, given its value to both software buyers and sellers. The unique nature of *PC Disk*—a diskette—removes the last obstacle to realizing the benefits of software demos. Our readers can try new software at their own convenience, and at no additional cost. Software sellers know they are reaching a highly receptive market at an affordable price. It is the perfect marriage of message and medium. We thank Microsoft for their pioneering spirit, and look forward to more of such offerings in future issues of *PC Disk*.



CONTENTS

DEPARTMENTS

TERMS AND CONDITIONS

LIMITED WARRANTY

In the unlikely event that the disk contained in this issue is defective in manufacture, return it within 30 days after purchase or receipt to *PC Disk*, P.O. Box 5930, Cherry Hill, NJ 08034, and it will be replaced at no charge. Merchandise for exchange must be accompanied by proof of purchase or a subscription label.

The publisher shall have no liability or responsibility to any purchaser or subscriber or to any other person or entity with respect to any liability, loss or damage directly or indirectly arising out of the use of the disk and the programs contained in this issue, including, but not limited to, any loss of business or any other incidental or consequential damages.

COPYRIGHT

A LIBRARY OF SOFTWARE FOR THE **IBM PC** & **PCjr**

PC DISK

PUBLISHER
André M.J. Van Hattum

EDITORIAL
Editor: Morris L. Effron
Software Editor: Dale Benzer
Managing Editor: Bohdan "Bo" Shmorhay
Documentation Editor: Bruce L. Murphy
Copyeditor: Laura Ninger
Assistant Editors: Anne Condon, Joe Dauben, Matt Dolph, Dean Hannotte, Doug Henwood, Evelyn Leon, Thomas Murphy
Production Coordinator: Eve Hinderer
Editorial Secretary: Lillian Pearsall

ART & DESIGN
Cristina Botta, Borys Patchowsky
Susan Molloy, Thomas Stvan.

CIRCULATION
Subscription Director: Chet Klimuszko
Media Manager: Ann Pichette
Direct Mail Manager: Eric A. Bernhard
Renewal and Billing Manager: Shane Boel
Special Subscription Sales: Daniel Rosensweig

CONSUMER COMPUTER AND ELECTRONICS MAGAZINE DIVISION
President: Larry Sporn
Vice President, Marketing: J. Scott Briggs
Vice President, Circulation: Carole Mandel
Vice President, General Manager: Eileen G. Markowitz
Vice President, Licensing and Special Projects: Jerry Schneider
Vice President, Creative Services: Herbert Stern
Creative Director: Peter J. Blank
Editorial Director: Jonathan D. Lazarus
Marketing Manager: Ronni Sonnenberg

ZIFF-DAVIS PUBLISHING COMPANY
President: Richard P. Friese *President, Consumer Magazine Division:* Albert S. Traina *Executive Vice President, Marketing and Circulation:* Paul Chook *Senior Vice Presidents:* Philip T. Heffernan, Sidney Holtz, Edward D. Muhlfeld, Philip Sine *Vice Presidents:* Baird Davis, George Morrissey, Rory Parisi, William L. Phillips *Treasurer:* Selwyn Taubman *Secretary:* Bertram A. Abrams
PC Disk is published 9 times per year, by Ziff-Davis Publishing Company, One Park Avenue, New York, N.Y. 10016. Subscriptions are available at $149.00 for 6 issues. POSTMASTER: send address changes to *PC Disk*, P.O. Box 5930, Cherry Hill, NJ 08034
Editorial and Business Office: One Park Ave., New York, NY 10016. Editorial (212) 725-7770. For subscription inquiries and service write to *PC Disk*, P.O. Box 5930, Cherry Hill, NJ 08034
PC Disk is an independent journal, not affiliated in any way with International Business Machines Corporation. IBM is a registered trademark of International Business Machines Corp.
ISBN # 087165-108-4 PC DISK VOLUME 1 NUMBER 8

TECHNICAL PREFACE

To help our readers make the most of PC Disk Magazine, *we would like to provide some background information concerning the editorial diskette, the accompanying manual, and how to use both. We don't expect all of the following topics to be of interest to all our readers. Nevertheless we prefer to err on the side of comprehensive support, rather than leave any of our readers confused or bewildered. So we encourage everyone to at least skim this section to assure a solid background for the use of* PC Disk Magazine.

USAGE REQUIREMENTS

PC Disk Magazine has been designed for use on an IBM Personal Computer with a minimal set of hardware components: a keyboard, a monitor, and the computer itself. Running DOS 1.1, a minimum of 64K of main memory is required. Under DOS 2.0 and DOS 2.1, a minimum of 128K is required. The display unit can be a monochrome display adapter and monitor, or the color graphics display adapter with either a color monitor, a black and white monitor, or an RF Modulator and TV set. The computer itself can be the PC, the PC-XT or the PCjr. For the PCjr we recommend the use of a monitor rather than a TV set, since most of our software is written for an 80-column display. However, a TV set does provide a workable display.

These three pieces of equipment are all you need to run the majority of *PC Disk Magazine* software. Wherever possible we try to make the use of any other hardware optional. So, for example, many of the programs will generate printed output, but a printer is not required to use them. Occasionally, however, due to the nature of a program or its design, a particular piece of equipment will be necessary. When a program requires a piece of equipment not in the minimum configuration stated above, this component will be listed as a "Special Requirement" on the program's title page in this manual.

In regard to software, all *PC Disk Magazine* programs are designed to run under DOS 1.1, DOS 2.0 and DOS 2.1. Furthermore, all BASIC programs in the magazine are designed to run under Microsoft's Advanced BASIC. Neither DOS

nor Advanced BASIC are provided on the *PC Disk Magazine* diskette; they must be acquired separately. As a rule, these are the only outside software elements you will need to use *PC Disk Magazine*. We will occasionally publish a program which uses some additional, publicly available software product. Any such additional software will be listed as a "Special Requirement" on the program's title page in this manual.

A closing remark on this topic is not so much a requirement as a recommendation. We recommend that you make a copy of your *PC Disk Magazine* diskette to work with, and save the original as a backup. In some cases, you will have to make a copy of the program in order to use it. The reason is that some programs create additional files as they run, and these files must be stored on diskette as well. You may have noticed that your *PC Disk Magazine* diskette is write-protected. Thus it cannot receive these additional files. So a separate, working copy is needed. These situations will be explicitly mentioned in the manual. In general though, where the manual refers to "your *PC Disk Magazine* diskette" you should read "your working copy of the *PC Disk Magazine* diskette."

THE IBM PC KEYBOARD

In *PC Disk Magazine* we have tried to make our instructions as clear as possible by the consistent use of special key symbols. In addition to all the common typewriter keys, which we print as they would appear when typed, the IBM PC keyboard has a number of special keys. We have designed symbols for these keys, which are intended to resemble as much as possible the keys themselves. Since these symbols are used extensively throughout the instructions, we felt the road map and glossary on the following page would help you, our reader, get any needed bearings.

ATTENTION PCjr OWNERS:

The instructions in our manual are written for the PC keyboard. Section 4 of your "Guide to Operations" manual for the IBM PCjr provides complete information on how to translate PC keystrokes to PCjr keystrokes. Please refer to this section for guidance in following our operating instructions.

TEXT CONVENTIONS

Most of the textual conventions of this manual are fairly obvious. The use of special key symbols has been covered. The use of special key names in the narrative text has been discussed. That leaves two brief additional remarks concerning command lines.

The lines set apart from the narrative text are commands that should be typed in exactly as they appear. When two key symbols appear immediately next to each other in such a command line, they should be pressed simultaneously. For example:

means press the Shift key and the Print Screen key simultaneously, thereby printing a copy of the current screen on your printer.

There is one exception to typing in command lines exactly as they appear. When a command includes a phrase such as "somename" or "programname" or "yourfile" you should replace that phrase (but not any punctuation) with a valid filename of your choice when you enter the command.

F6 THE FUNCTION KEYS

There are ten special keys called function keys located at the far left of the keyboard. They are numbered from F1 to F10. This stands for Function One, Function Two etc. These keys are often used to make single keystroke choices or commands.

Esc THE ESCAPE KEY

The Escape key is used most often for exactly what its name implies, to escape (exit) from various functions and processes.

Ctrl THE CONTROL KEY

This key is always used in conjunction with another key by pressing this key and the other key simultaneously. The purposes of the Control key vary widely depending on the application program.

|← →| THE TAB KEY

This key is commonly used for horizontal tabbing in text files. It is sometimes used by programs to allow rapid cursor movement during full-screen data entry.

← THE BACKSPACE KEY

The Backspace key is used to correct typing errors. By simply pressing the key, the preceding character is erased and a new character can be entered.

⇧ THE SHIFT KEY

The Shift key is actually located on each side of the keyboard. It is used in conjunction with other keys to capitalize letters, get special symbols like: ! @ # $ % * () and other special functions.

PrtSc * THE PRINT SCREEN KEY

This key is used with the Shift key to get a printout of exactly what is on the screen. In computer lingo this is called a screen dump, a dump of all the information on the screen to the printer. In *PC Disk Magazine* we also refer to this capability as "The IBM Print Screen Facility."

↵ THE ENTER KEY

This is the most frequently used key on the keyboard. Almost every time you need to give information to the computer, you have to press this key to ENTER that information. This key can also be thought of as the carriage return, since it works similarly to the RETURN key on a typewriter.

Num Lock THE NUM LOCK KEY

This key toggles between calculator mode and cursor control mode. To change from one mode to another press the NUM LOCK key once.

2 ↓ 4 ← 6 → 8 ↑ THE CURSOR CONTROL KEYS

These are the arrows that point up, down, and to each side. If these keys are not functional, press the NUM LOCK key once. These keys control cursor movement within some *PC Disk Magazine* programs. They will move the cursor in the direction of the arrow.

0 Ins . Del THE INSERT AND DELETE KEYS

These keys really mean the INSERT and DELETE keys. And that is exactly how they are used. INS is used to insert new information and DEL is used to delete unwanted information. They are commonly used when editing BASIC programs, and can often be used when running BASIC programs as well.

Caps Lock THE CAPS LOCK KEY

This key is used to save you from having to hold the shift key down all the time to get capital letters.

Ctrl Scroll Lock THE CONTROL AND SCROLL LOCK KEYS

This key combination deserves special mention because of its importance in BASIC, the language of most *PC Disk Magazine* software. These keys used together will interrupt the processing of any BASIC program. The keys should be used with caution because some interruptions can require you to start an entire procedure from the beginning.

TERMINOLOGY

In the preceding section we identified the special key symbols used in this manual, and gave a name to each one. For example:

↵

is called the Enter key. In our instructional narrative, it sometimes makes more sense to refer to a special key by its name rather than its symbol. Thus the key names in the preceding section are also special terms for the purposes of this manual. Familiarize yourself with the names to facilitate your use of the manual, and refer to the preceding section as a glossary of key names when necessary.

In addition to the key names, a few other terms and phrases are used in this manual that may be unfamiliar to you.

We commonly speak of putting a diskette in the "default drive." This may seem like a needlessly vague phrase. After all, we know a diskette drive always has a one letter identifier associated with it, so why not refer explicitly to that letter? The problem with using an explicit letter reference is that it can create confusion about what exactly you must do. In other words, operationally it does not matter whether you put the diskette in the A Drive, the B Drive or even the C Drive (if you have a third diskette drive). What matters is that you put the diskette *in the drive that is currently active*, i.e. the drive whose letter prompt currently appears on the screen. This is your "default drive" because any disk command without a drive letter will look at the diskette in this active drive. So when you put a diskette in the "default drive," you can then issue commands referencing that diskette without the use of letter identifiers.

Every start-up procedure for a BASIC program requires you to "Load Advanced BASIC into your PC." To run a *PC Disk Magazine* BASIC program, the BASIC Interpreter must be up and running on your machine—you must be "in BASIC." BASIC is really a program like any other. To start it you must load it from a disk into your PC and start it running. This is precisely what happens when you put your DOS diskette (or any diskette with the file BASICA.COM) in the default drive and type:

BASICA ↵

By so doing, you "Load Advanced BASIC into your PC."

PUBLISHING PHILOSOPHY

PC Disk Magazine believes in giving you the most for your software dollar. Consequently, wherever possible we publish BASIC programs in the original source code, which allows you to list and copy these programs in their entirety. Furthermore, our editorial diskette itself is not copy-protected in any way, making it possible for you to produce work copies of our software. We at *PC Disk Magazine* choose neither to copy-protect our offerings, nor to make our program code inaccessible—a rather sharp departure from traditional software distribution.

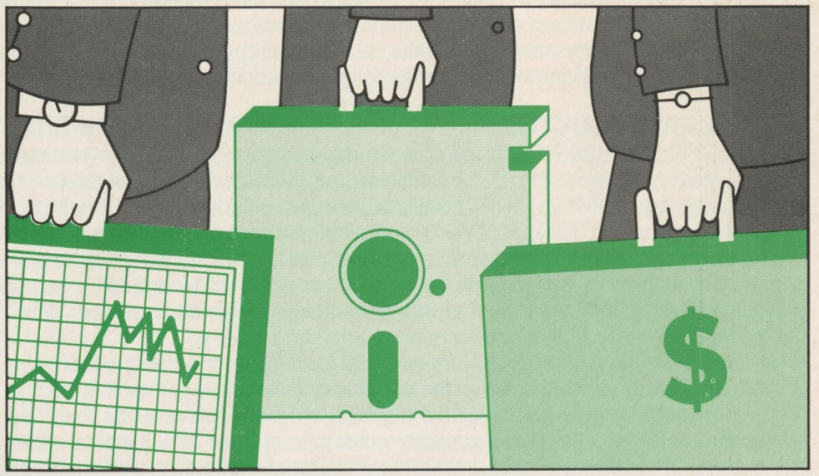
WHY?

The most important reason behind our decision is our desire to make *PC Disk Magazine* of the greatest possible use to you. We want *PC Disk Magazine* to expand the utility of your microcomputer—the reason you bought the machine in the first place. This desire involves several considerations. First, we want you to be able to learn from these programs. We believe that many of our programs can serve as models of good programming practice, and also provide routines of general use. To derive this benefit you must be able to study the source code in order to understand the design concepts and programming techniques employed. Second, we want you to be able to adapt these programs to your own needs. You may choose to modify some programs, or expand them, or include portions of them in programs of your own. Once again, you need the source code to do this. The third consideration is that to have the flexibility to use *PC Disk Magazine* most extensively, to modify and expand programs, to create different versions and to experiment, you must be able to make copies of *PC Disk Magazine* materials. Thus, our objective—to establish the most productive relationship possible between this magazine and its readers—was the most compelling reason to publish copy-able source code.

Another reason worth mentioning is space. The same program requires 10 to 50 percent less storage space on our editorial diskette in source code than it would in compiled BASIC. So publishing programs in source code lets us give you more software in the fixed amount of diskette space available.

Now that you understand the reasons for our decision, we hope you will minimize our risk by honoring the legal copy restrictions that apply to *PC Disk Magazine*. Use the magazine as extensively as you like for yourself, but do not give listings or copies of our software materials to others. Be aware that the contents of *PC Disk Magazine* are copyrighted private property. Your technical freedom to copy these materials implies no legal right to distribute them. We ask that you act responsibly in your use of *PC Disk Magazine* and not abuse the spirit of open exchange. For our part, we will continue to make our material as useful to you as possible, with the expectation that, properly understood, this policy will best serve you our readers.

BUSINESS



STATISTICS LIBRARY 1

Designed by Dale Benzer
Programmed by Peter Schlaifer

Special Requirements: DOS 2.0, min. 128K (printer optional)
Files Used: STATLIB1.BAS
STEDIT.BAS
ST.TXT

If you have ever had to deal with a mass of data and needed to figure averages, means, statistical deviations, or plot the results graphically, then this program is for you. Now you can quickly and conveniently compute a variety of statistical functions. STATISTICS LIBRARY 1 *is the first in a series of* PC Disk Magazine *statistical packages that will enable you to build your own statistics library.*

This versatile program has unlimited potential. If you are a teacher, for example, and need to analyze class records, test results, etc., you can plot the spread of grades on an examination; the mean, median, mode, and so on. If instead you are interested in business applications, this package can help create a database that you can update and analyze for any number of statistical combinations. If you want to find out how many salespeople in a particular region have sold more than 10,000 widgets to hardware stores in Peoria, this package can do the job. After you set the variables and determine which data you wish to cross-tabulate for comparisons, you can sit back and let the computer be your statistician.

BACKGROUND

STATISTICS LIBRARY is a statistical package which cross-tabulates data, produces frequency tables, histograms, and scatterplots, and computes means, me-

dians, ranges, and standard deviations. You can define a list of up to 105 variables for statistical analysis and cross-tabulation. You may enter a different set of data for each variable for as many cases as you have in your sample. For example, if you have a number of questionnaires ("cases") including such information as sex, age, income, etc., this program tells you how to label individual variables (like sex, age, income), and then enter the data for each questionnaire. Then, by defining various subsets of variables you want to correlate (age and income, for example, or only data from all respondents over the age of 30), you can cross-tabulate your data and create frequency charts, histograms (bar graphs), and scatterplots. This program will also compute ranges and standard deviations.

PROGRAM STRUCTURE

The *STATISTICS LIBRARY 1* program is actually two programs in one. The first is the data-input program (STEDIT.BAS) that is used to establish your database. The second program (STATLIB1.BAS) contains the statistical routines used in this first installment of the *STATISTICS LIBRARY* series. Once you run the initial sequence and input all the data for your sample, you can move on to the second program and perform your analyses. The program has been split into two sections so that you can return to your original data sample and run repeated analyses without having to wade through the input sequence each time.

Except for the Main Menu, the screen format is standard for both program segments. The Data Screen consists of the variables, data values, subsets, and other pertinent information. Below the Data Screen is a list of the relevant command options and a command line that prompts you for your entries. All your data entries and command choices are made in this line. The Data Screen above will automatically reflect these entries.

START-UP

In order to use *STATISTICS LIBRARY 1*, you must transfer the necessary programs from the *PC Disk Magazine* diskette to a formatted diskette that has room for the additional files which the program generates. To do so, place your *PC Disk Magazine* diskette into your default drive and type:

```
COPY STATLIB1.BAS B: ↵
```

Repeat this procedure for the other files the program requires, STEDIT.BAS and ST.BAS. To run *STATISTICS LIBRARY 1*, first load Advanced BASIC into your PC by putting a system disk in your default drive and typing:

```
BASICA ↵
```

then place your work diskette into your default drive and type:

```
RUN "STEDIT ↵
```

This will start the data-entry portion of the *STATISTICS LIBRARY 1* package. Once you have established your database, you will want to run the statistical analysis portion of the program. To do this, return to Advanced BASIC and type:

```
RUN "STATLIB1 ↵
```

DATA-INPUT PROGRAM

If you are using *STATISTICS LIBRARY 1* for the first time, you must construct a database by defining variables and entering data for each case in your sample. Run the data-input portion of the program as detailed in the Start-Up section. The Main Menu displays a list of data-entry options, along with a submenu that describes some global commands that are active throughout all program operations.

CREATING A NEW FILE

The first thing you must do is create a file that will hold your data. Choose option "C" to create a file, then specify a filename. This name can be up to eight characters in length. Next you will be prompted to enter a file description of up to 25 characters. This feature is provided in case you want to enter some descriptive information about your data sample. The Data Screen, available options, and the command line are then displayed. Above the Data Screen are the filename and the case number, which the program calculates automatically.

DEFINING VARIABLES

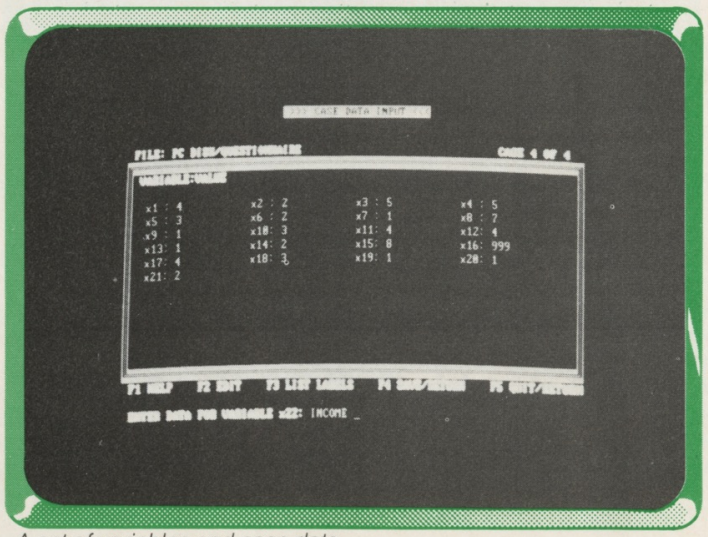
After opening a new file, choose option "A" to begin entering variable names. The command line will display the first variable, x1. Enter a variable name of eight characters or fewer, then press the Enter key. Continue this procedure until you have entered the variable names for your entire sample.

Suppose your database consists of 23 variables, including the sex, age, income, occupation, etc. of respondents to a marketing questionnaire. You would define your variables as x1=sex, x2=age, x3=income, etc. until all your variables have been entered.

The additional options listed below the Data Input Screen allow you to modify your variables. You may insert, delete, or reenter variables. Each of these options will prompt you for a variable number, then display that variable in the command line and allow you to make your change. When you insert and delete variables, the list is updated and the variables are renumbered. When you reenter a variable, the numbering remains unchanged. If at any point you run into a problem, select option "H" to display the Help Screen. When you have finished entering your variables, be sure to save them to disk with the Save option.

ENTERING CASE DATA

STATISTICS LIBRARY 1 is constructed in such a way that it only accepts data in



A set of variables and case data

numerical form, up to a maximum of four digits. This may require that your information be coded before you begin entering data. This simply means that you must assign numerical values to the information you wish to enter. For example, if your first variable were sex, you might assign "1" for males and "2" for females. If age were your second variable, you might want to assign numbers to specific ranges, for example, "1" might designate anyone less than 20 years of age, "2" those between 20 and 25, "3" all those between 25 and 30, etc.

To begin entering data, return to the Main Menu, then press "I" to input case data. The variables you previously specified for this file will be displayed along with the filename and case number. Variable x1 will be displayed in the command line waiting for data input.

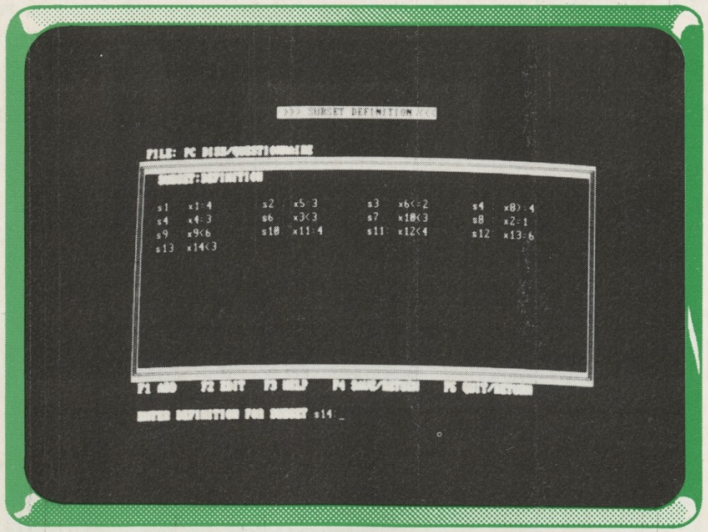
After you enter data for the first variable, press the Enter key and the variable x2 will appear in the command line. Enter data for this and all subsequent variables. If you want to change an entry, choose option "R", specify the variable you want to change, and enter the new data value. After entering data for one case, move on to the next case by choosing option "N". This option will first prompt you to save the previous case and then allow you to enter data for the next case. The program keeps track of the total number of cases as they are entered, and adjusts the "Case x of x" information accordingly.

When you have finished entering data for all your cases, choose option "S" to save your data and return to the Main Menu. The remaining options on the Main Menu allow you to delete case data, reenter data for a particular case, print your data file, view a Help Screen, or exit to DOS.

DEFINING SUBSETS

Once your database is established, you are ready to designate the variables you want *STATISTICS LIBRARY 1* to tabulate. If you want to perform statistical tests on all the variables, you may proceed to the statistical routines at this point. If, however, you wish to examine specific cases, for example those in which the respondents were male, you must define a subset before doing any statistical tests.

First select option "S" at the Main Menu of STEDIT.BAS. The program will then



The Subset Definition Screen

prompt you to begin defining subsets. Suppose you do want to examine all of the information in your database for males. You could then define subset s1 as the set of males represented in your database (e.g. those who have responded to the questionnaire and who are male). Since we have previously defined the variable for the sex of a given case as x1, and coded the variable for male as 1, then subset s1 would be defined as x1=1. After you type this and press the Enter key, the cursor will move on to s2. If you wanted subset s2 to define all females represented in the database, you would type x1=2, and press the Enter key.

The additional options listed below the subsets allow you to insert, delete, or reenter your subsets. Use the same procedure that you used to modify your variables (see Defining Variables above).

THE REJECT CASE OPTION

Defining a subset will cause the program to reject a case in which the subset is not true. A further option can be used that will reject a case in which any variable has a value of 0. Use this option to eliminate incomplete cases.

To turn the Reject option on, simply type "R" at the command line when you define your subsets. The program will display "REJ IF 0" as an entry on the subset screen.

THE RELATIONS <, >, <=, >=

In addition to the relation for equality or identity (=), you can use other logical relations when defining subsets. These are "<" ("Less than"), ">" ("Greater than"), "<=" ("Less than or equal to"), and ">=" ("Greater than or equal to"). Suppose, for example, you want to identify all those people in your database whose incomes are greater than $40,000. If you had coded the range of incomes between $35-40,000 as a "3", you could then define subset s3 as x3 (the variable for income) >= 3 (remember that all data is entered numerically and can be no more than four digits long).

SELECTING SUBSETS

When analyzing your data, you may want to limit the case data under consideration by using more than one subset definition. Suppose you wanted to find the number of males between the ages of 25 and 30 with incomes of more than $30,000. This requires you to define three subsets: one that identifies all the males, one identifying those between the ages of 25 and 30, and one identifying those earning more than $30,000 per year. Subset s1 would be defined as s1: x1 = 1, where "x1" is the variable for sex and "1" is the code for male. Subset x2 would be defined as s2: x2 <= 3, where "x2" is the variable for age and "3" is the code for 25-30 years old. Subset x3 would be defined as s3: x3 >= 3, where "x3" is the variable for income and "3" is the code for $25,000-30,000.

DERIVED VARIABLES

While defining a subset file will allow you to work with selected data from your cases, the derived variable option allows you to define a new variable in terms of existing variables or constants. For example, you might want to set up a new variable, x4, that will hold the values of existing variables x1 + x2. All operations, including subset definitions, can be performed using this new variable. To define a derived variable, load the STATLIB1.BAS file, then press "D" at the Main Menu. Proceed as if you were entering a subset, the only difference being that the label will appear as d1 rather than s1. The program will include these new variables in the data file.

STATISTICAL ANALYSIS

After defining your subsets, return to the Main Menu. You are now ready to put *STATISTICS LIBRARY 1* to work analyzing your database. Select option "T" (Sample Description) to access the following statistical routines: Frequency Table, Histogram, and Scatterplot; Mean, Median, Range, and Standard Deviation. First enter the filename of the database you want to analyze, the subset filename (if you choose), then select one of the statistical options.

FREQUENCY TABLE

Suppose you want a description of the frequency of males in the database by age group, from 20 to 40, broken down into five-year intervals. Before *STATISTICS LIBRARY 1* can print a frequency table, the program needs to know what variable you are interested in analyzing, the low and high ends of the desired statistical range, the number of intervals, and the step values you want computed. Select the Frequency Table option, and the corresponding screen will appear.

First you must specify the subset(s) to be analyzed. If you want to examine all cases, enter "All". If you are only interested in a particular subset, indicate which one.

The cursor appears at the command line so that you can enter the variable to be examined for frequency. In this case, the variable that interests us is age, so enter x2. Since we want to know the number of males in the database between the ages of 20 and 40, enter 20 on the "Low start" command line, and 40 on the "High end" line. Since we also want the data broken down into five-year intervals, and there are four such intervals between 20 and 40, enter a 4 on the "Number of intervals" line, and a 5 on the "Step value" line. Press the Enter key to process this information.

The resulting screen lists the filename, subset file, variable, low start and high end values, and the count and percentage for each step. A cumulative count and percentage are also kept. This is a good way to see how close each step brings you to 100%.

To obtain a hard copy of your frequency table or any of the other statistics displays, press:

HISTOGRAM

If you prefer to display statistical frequencies in a bar chart rather than in the form of a frequency table, select the Histogram option. The command lines at the bottom of the screen are identical to those used in generating a Frequency Table. Move the cursor to the various command lines at the bottom of the screen and enter the appropriate subset, variable, and interval information. The program processes your data and displays a horizontal bar graph with the range of 0 to 100%. Each interval has its own bar composed of asterisk symbols.

SCATTERPLOT DIAGRAM

Sometimes it may be useful to plot one variable against another. Select the Scatterplot option from the Sample Description Menu. The command lines at the bottom of the screen prompt you to enter the variable subscripts for the vertical and horizontal axes, as well as the subset.

If you wanted to correlate income and age, you would enter x2 for the age variable and x3 for income. If you do not want to limit the scatterplot to any of the subsets, type "None" and press the Enter key to execute the scatterplot.

If you want to produce another scatterplot for a different combination of variables, press "Y" at the prompt. Once you have produced all the scatterplots you want, press the Escape key to return to the previous options and the other statistical choices.

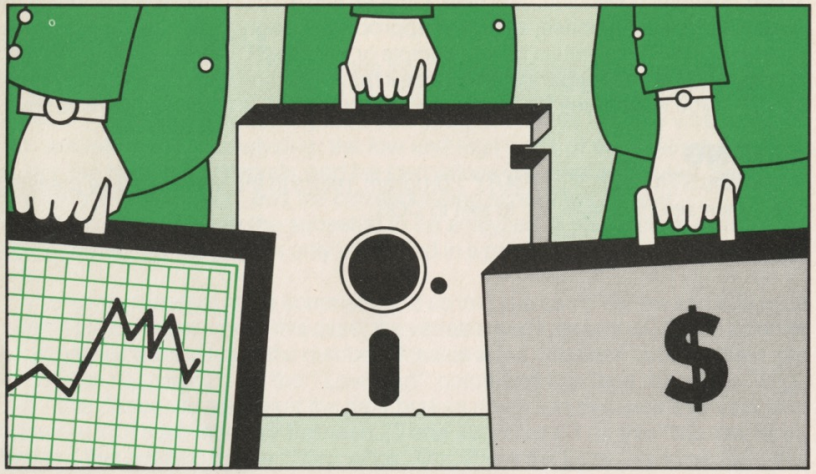
STATISTICAL FUNCTIONS

The remaining option of the *STATISTICS LIBRARY 1* program in this issue of *PC Disk Magazine* computes a number of basic statistical functions. To obtain the Mean, Standard Deviation, Smallest, Largest, and Median values of the ranges of variables in your database, select the appropriate option. These values are calculated for all the case values in your sample or the subset you specify. The Median computation is done separately because it requires a sort and consequently takes longer. If you want to see the Median values, press "Y" in response to the prompt.

EXITING

To exit the *STATISTICS LIBRARY 1* program, return to the Main Menu by pressing "M", then select option "X" to exit to DOS.

BUSINESS



SPREADSHEET CONSOLIDATOR

By Robert E. Josephson

Special Requirements: None
Files Used: SPREAD.BAS
DIR.BAS

A certain company is on the forefront of technology. Its managers saw that personal computers were the wave of the future and they made the big investment. Now, each division has an IBM PC. Disks are flying about everywhere, telephone lines crackle with the song of the modem, and every week the main office receives detailed financial reports from each division. It seems that everyone is happy.

But the system is not flawless. There is still one unfortunate person who spends hours of valuable computer time entering this information on the main office's spreadsheet program. Each week, this employee must reconcile reports that were formatted with several different spreadsheet programs. No doubt he or she wishes that there were a program that could combine the data into one file.

SPREADSHEET CONSOLIDATOR *does just that. The program combines spreadsheet files, which have been saved in a standard "DIF" format, into a new DIF file according to a format that you select. Now the worker who used to type in all the information manually can use this program and take the rest of the week off!*

BACKGROUND

DIF, a trademark of Software Arts, Inc., is a standard format for spreadsheet data storage. Information saved in DIF files is transportable among many popular spreadsheet programs such as VisiCalc, Lotus 1-2-3, etc., though not among machines. The standard format enables *SPREADSHEET CONSOLIDATOR* to read files from several programs and consolidate them into one file. The combined data is arranged by title and label, according to a pattern set up in a master file. This master file can be created in the format of any DIF-supporting spreadsheet program.

START-UP

To use *SPREADSHEET CONSOLIDATOR*, first create your master file, which then serves as the pattern for arranging the combined data from the subsidiary files. The master file need contain only those titles, labels, and captions that are likely to appear in the subsidiary files. The master file can contain data or it can simply be a template for organizing your files. Since *SPREADSHEET CONSOLIDATOR* ignores capitalization and extra spaces, it is not crucial to match exactly the titles in the master file to the titles in the subsidiary files. For example, figures for "Sales" and "sales" would be consolidated under the master heading "SALES". It is also possible to use the labels of one of the subsidiary files as the master pattern. The only restriction is that your master file and subsidiary files be resident on the same diskette.

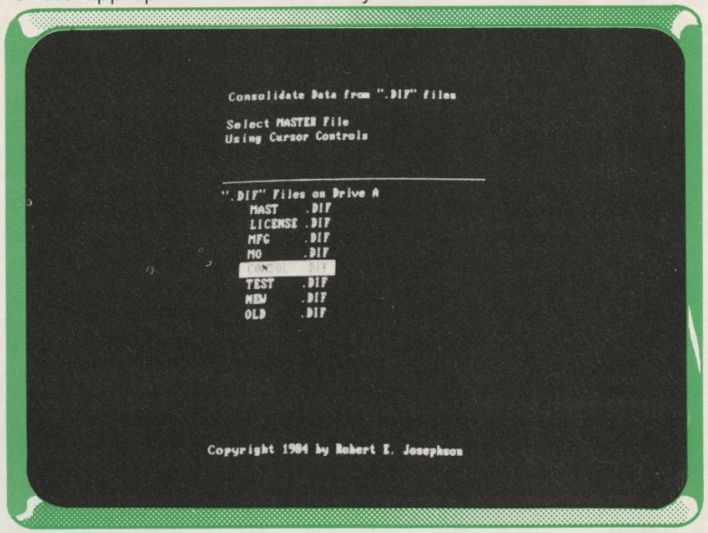
To start *SPREADSHEET CONSOLIDATOR*, you must be in Advanced BASIC. Make sure your work diskette contains the files SPREAD.BAS and DIR.BAS (this latter file is a machine language subroutine which reads the diskette directory). Type:

```
RUN "SPREAD ↵
```

You will be shown the Title Screen and then asked to choose a master file.

CHOOSING THE FILES

SPREADSHEET CONSOLIDATOR first asks for the letter of the drive which holds the master file disk. Use the cursor control keys to move the highlighted box to the letter of the appropriate drive. You may remove the disk which contains the



Choose a master file

SPREADSHEET CONSOLIDATOR programs—they have already been loaded. Press the Enter key after selecting the appropriate drive.

SPREADSHEET CONSOLIDATOR then displays all the files on the disk in that drive that have the .DIF extension. (This extension is appended to any files that were created in the DIF format.) Use the vertical cursor control keys to move the highlighting box to the name of the master file. Press Enter to select this file. (If you choose a file which for some reason cannot be opened, the program gives you an error message and prompts you to select another file.)

After choosing a file, indicate whether this file has been stored in DIF format by row or by column. The program must have this information in order to read a DIF file properly, so type "R" or "C" accordingly. The program then takes a moment to process this file and labels it on the displayed directory with the marking "*M*".

Once you choose a master file, the program follows the same file selection procedure for the subsidiary files. Each time you select a subsidiary file and indicate whether it has been stored by row or column, it is processed and loaded into memory. Thereafter, the mark "***" will appear next to the filename on the DIF file directory. The number of files you can process at one time is limited only by your computer's memory space, and the order in which you select the subsidiary files has no bearing on the consolidation process.

When you have chosen all of your subsidiary files, press:

E ↵

CONSOLIDATION

At this point, *SPREADSHEET CONSOLIDATOR* prompts you for the name of the file where you wish to place the consolidated data. This file automatically receives the .DIF extension. If you wish to write this file to somewhere other than the current disk drive, prefix the filename with a drive name. The consolidated file will be stored by row.

After you enter the output filename, all your file specifications appear for verification (i.e. master file, subsidiary files, and output file). If you want to change any of these entries, press "N" and you will return to the master file prompt. Once you confirm the filenames, processing begins.

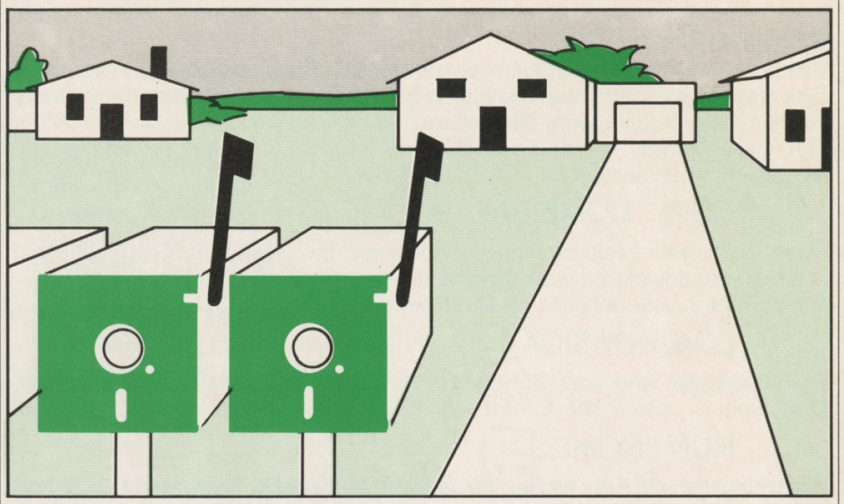
If *SPREADSHEET CONSOLIDATOR* finds a title or caption in a subsidiary file that is not included in the master file, it displays a message saying that the label cannot be matched. You then have the option to include or omit that line in the consolidated file.

SUBMISSIONS

If you have developed professional software for the IBM PC that you feel would be suitable for publication in *PC Disk*, we would like to know about it. Publishing is no longer just for poets and prosaists, but now it's for programmers too. Ask for our Software Submission Plan by writing to:

PC Disk
Author Submissions
One Park Avenue
New York, N.Y. 10016

PERSONAL



HOME INVENTORY

By F1 Software

Special Requirements: None
Files Used: HOME.BAS
HOME.TXT (created by program)

Home security begins with an awareness of your possessions. Home-owners and apartment-dwellers usually undervalue the things they own and underestimate replacement costs. This happens because home furnishings accumulate gradually, and they become so familiar to you that they seem invisible. Only by making a careful inventory of your material possessions can you determine the appropriate amount of fire or theft insurance to carry. Recovering stolen possessions is even easier if you can provide the police with serial numbers and item descriptions. Ultimately, if possessions are unrecoverable due to fire or burglary, you need to accurately assess the uninsured tax loss or insurance claim.

With the HOME INVENTORY *program, you can realistically determine the value of the contents of your home, apartment, or office.* HOME INVENTORY *allows you to list your valuable possessions along with serial numbers, dates of purchase, replacement costs, and even sales tax and delivery charges. The program provides room-by-room totals and a grand total for the entire inventory. Items can be added or deleted with ease.*

BACKGROUND

All inventory items are stored in the HOME.TXT file which the program creates for you. You enter an inventory item by completing nine descriptive fields. These

fields help you to identify each of your possessions by location, serial number, purchase price, replacement cost, date of purchase, and any other charges associated with their purchase. The program sorts the inventory file into a room/item sequence, then provides a final report giving dollar values by room and an overall grand total. You can browse through the file, as well as add, change, or delete any record. You can specify search criteria to find a certain item, as well as choose the "help" option that explains all processing options and function keys.

START-UP

Before running *HOME INVENTORY*, transfer the necessary files from your *PC Disk Magazine* diskette onto a separate formatted diskette that has room for the inventory file which the program creates.

To make the copy, you must be in DOS. Put your *PC Disk Magazine* diskette into the default drive and type:

COPY HOME.BAS B: ↵

where "B" is your destination drive. On a single drive system, DOS prompts you to insert your destination work diskette. To run *HOME INVENTORY*, insert a system diskette in your default drive and type:

LOAD BASICA ↵

Then place your work copy of *HOME INVENTORY* in the default drive, and from DOS, type:

RUN "HOME ↵

The first screen prompts you for the name of the input file. If you are browsing or modifying an existing file, enter that filename. If you are creating the inventory file for the first time, you can either assign your own filename or use the program default name, HOME.TXT, by pressing the Enter key. If you use different files, you can create separate inventories for your home, office, summer house, etc.

When creating a new inventory file, the program returns a message saying: "File not found—can create new file: Continue, Retry, Abort (C, R, A)?" To create the new file, select "Continue" by pressing the Enter key.

PROGRAM OPTIONS

The Data Entry Screen offers four processing options. The Add option allows you to create or to place additional items in your inventory. The default option, Browse, permits you to display your inventory item by item. The Change option allows you to modify all or part of a record, while the Delete option lets you eliminate items from the inventory. It is important to note that the Add, Change, and Delete options only take effect after the F10 key is pressed.

HOME INVENTORY uses all ten function keys:

F1 —functions as the HELP key.
F2 —closes the file and ends the session.
F3 —erases data from the cursor to the end of the field.
F4 —restores all data fields with their previous contents.
F5 —finds the first appearance of a specified item.
F6 —finds the next appearance of the specified item.
F7 —sorts the file in room/item order.
F8 —creates a report of the room content values and a grand total.
F9 —displays the previous record in the file.
F10—displays the next record when Browsing, or finalizes the entry when Adding, Changing, or Deleting.

CREATING YOUR HOME INVENTORY

To create your inventory, select the Add processing option by typing:

A ↵

The cursor appears at the "Room" field, since the program sorts first by room and then by item. *HOME INVENTORY* prompts you for data by moving the cursor down field by field. Once the data for every field has been entered, press:

↵

When entering price information, round up to the nearest whole dollar. The program will not accept numbers to the right of the decimal point. The fields, and the maximum number of characters for each, are the following:

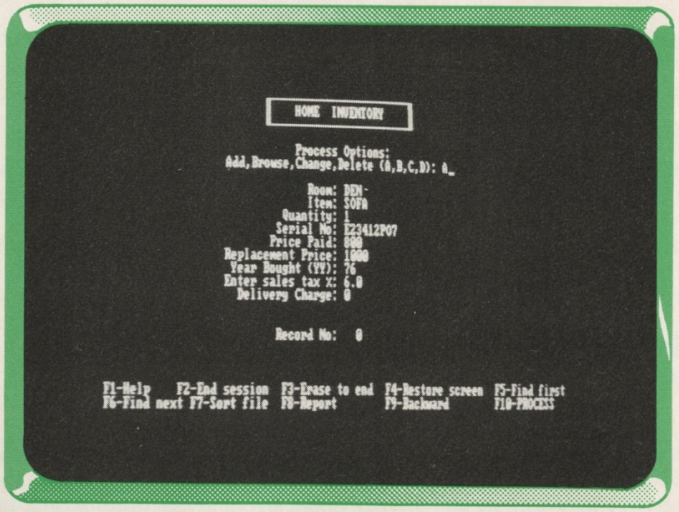
Room: (8 characters)
Item: (35 characters)
Quantity: (4 characters)
Serial Number: (15 characters)
Price Paid: (6 characters)
Replacement Price: (6 characters)
Year Bought (YY): (2 characters)
Enter Sales Tax (%): (4 characters)
Delivery Charge: (4 characters)

To finalize the entry and record the item in the inventory file, press:

F10

If you do not press the F10 key, the data in the screen will be deleted automatically. Once the item is successfully recorded, the program signifies that it has been "Added". After each record is added, you must again select a processing option to continue.

When you have entered all the items for each room in your inventory, sort the file in room/item sequence by pressing key F7. The sort step takes only a short



List your inventory by room

time, during which the program notifies you that there is a "Sort in progress". When the sort is done, the program returns you to the first record in the new sequence. You can now obtain a comprehensive inventory report, with totals by room and a grand total, by pressing key F8.

HOME INVENTORY next asks if you would like the report printed. The program will direct the report to the screen or the printer, or if no report is desired, it will let you quit. After you make your choice, press the Enter key.

FINDING A SPECIFIC ITEM

You can locate specific items with the Browse processing option, which is the default value on the Data Entry Screen. To pass forward through the file, record by record, press:

F10

To step backward through each record, press:

F9

If this seems tedious, you may locate the record by pressing the F5 key, entering the first few characters of the item description in order to identify it, and then pressing the Enter key. When a specified item cannot be found, the program informs you that it is "Not found". If you want to find other similar items in the file, you can display each successive occurrence by pressing:

F6

CHANGING A RECORD

Should any records need to be changed, first find the record, so that it appears on the Data Entry Screen, then select the Change option from the Data Entry Screen by typing:

C ↵

After you make each change in the field, press the Enter key again. To finalize and record these changes, you must then press:

F10

The program will confirm that the changes have been recorded.

DELETING A RECORD

To delete records, first find the desired record so that it appears on the Data Entry Screen, and then select the Delete option from the Data Entry Screen by typing:

D ↵

As with the Change option, you must finalize the deletion by pressing key F10. The program will verify that the deletion has occurred.

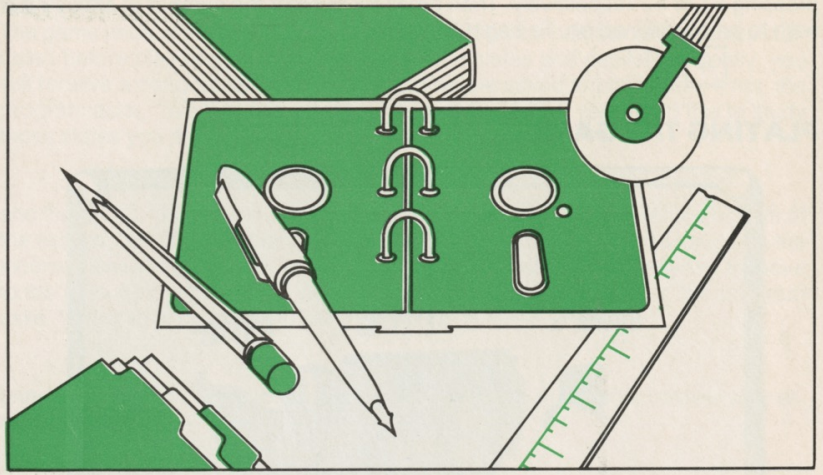
EXITING

When you have finished your inventory, press:

F2

to end the session. After verifying your intentions with a prompt, the program automatically creates a backup file. For complete security, it is advisable to keep a hard copy of your inventory somewhere outside your home, perhaps in a safe deposit box. Don't forget to include your computer in the inventory!

EDUCATION



STATES AND CAPITALS

By J. Edward Volkstorf, Jr.

Special Requirements: Color/Graphics Adapter
Files Used: STATES.BAS

Adults are not always equipped to answer the diverse questions of the inquisitive child. Questions like "Why is the sky blue?", "Are zebras black with white stripes or white with black stripes?", and "What's the capital of North Dakota?" not only put parents on the spot, but can stymie otherwise intelligent people.

While we cannot help you answer everything your child asks you, we can provide you with STATES AND CAPITALS, *an educational game that coaches you or your child on those puzzling geographical questions. Rather than saying "I don't know", or "Go ask your mother/father" when asked for a capital city, you can learn the answers from your IBM PC. The game is equally enjoyable and useful for the student.*

STATES AND CAPITALS *is a guessing game that tests your knowledge of the geography of the United States as well as your map-reading skills. The game's graphics outline the United States and the program awards you points for correctly identifying states and their capitals. If you still want to know the capital of North Dakota, keep reading.*

START-UP

To begin *STATES AND CAPITALS*, first put a system disk into your default drive and load Advanced BASIC by typing:

BASICA ↵

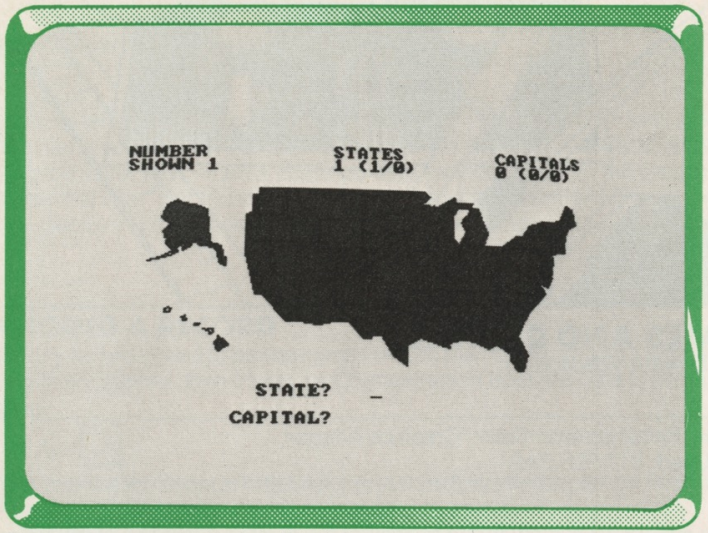
Next put your work copy of the *PC Disk Magazine* diskette in the default drive and type:

```
RUN "STATES ↵
```

The Title Screen appears with a list of your options for playing the game. Depending on your monitor type, set the game for color or black and white graphics by pressing F1 or F2 respectively. Then press F3 to enter the number of states you wish to be questioned on. To begin play, press:

F9

PLAYING THE GAME



Can you guess which state this is?

STATES AND CAPITALS is an easy game if you've done your homework. The program displays a map of the United States and highlights a particular state. You have two chances to guess the name of the state, and you can use the Backspace key, the Insert key, the Delete key, the Escape key or the left and right cursor control keys to correct any typographical errors. If you answer correctly, the program comments, "Good! Your answer is correct". If you enter the right name but spell it incorrectly, the program displays the message: "Check your spelling and try again". If you make two incorrect guesses, the correct state name will be displayed on the screen.

You are then prompted to enter the capital of the highlighted state. You have the same opportunity to answer with the correct capital name. If you misspell the city's name, the program again tells you that you were correct except for the spelling, and lets you try again. When you enter the wrong city name, the cursor returns to the beginning of the response line and you are allowed to try again. Guessing both the state and capital correctly rewards you with the message, "Very good! Both answers are correct". After both the state and capital names have been entered, press any key to move on to the next question.

A score line at the top of the Game Screen keeps track of how many states you have been questioned on, how many states you identified correctly, and how many capitals you knew. The score line also indicates the number of states and capitals that you identified correctly on the first try and how many you got right on the second try. This is shown in the form (aa/bb) where "aa" is the number of correct answers given on the first try and "bb" is the number of correct answers given on the second try.

REVIEW

The game ends when you have been questioned on the number of states you indicated before play began. If you entered all the states and capitals correctly, you will receive a congratulatory message and will be asked to play again. If not, the program asks, "Want to see what you missed (Y/N)?" If you would like to study those states you did not know, press:

Y

and the program will display the states on the map, one at a time. At the bottom of the screen, the correct state and capital names will be displayed along with the incorrect answers you entered. After you study each state, press any key to move to the next state and capital. When you have finished your review, the program asks, "Want to play again (Y/N)?" Pressing:

Y

brings you back to the Title Screen. If instead you do not want to review your errors, press:

N

and you will immediately reach the prompt asking if you want to play again or quit.

EXITING

You may stop playing *STATES AND CAPITALS* at any time during the game by pressing:

F10

The program then displays the Summary Screen, and asks "Do you want to see what you missed (Y/N)?" Press:

N

to exit the program and return to Advanced BASIC.

SUBMISSION PROCEDURE

What sets *PC Disk* apart from most other publications is that we have a voracious appetite for software. That's why we developed the "Software Submission Plan". If you're interested in submitting a professional piece of that software you've developed either on your own or with others, let us know, and we'll send you the Submission Plan booklet. This booklet provides details on our submission procedure, as well as our technical requirements for the software we publish. We would like to give you an outline of our Submission Plan here in order to stimulate your imagination and your interest.

The Software Submission Plan provides you with an opportunity to profit directly from your work. Under the plan, *PC Disk* will pay you either a flat fee or a royalty on every issue sold that contains your software. Best of all, this opportunity comes without elaborate restrictions. Our desire is to license software for publication in the disk magazine format.

To explore the considerable opportunities for publication in *PC Disk*, write to us and ask for a Software Submission Plan booklet. The address is:

PC Disk
Author Submissions
One Park Avenue
New York, N.Y. 10016

Once you've received the Submission Plan, read it over carefully. If you feel your software fits the requirements set out in the plan, complete the enclosed Software Submission Agreement and return it to us. Do not send us any program code or documentation when you return the Submission Agreement.

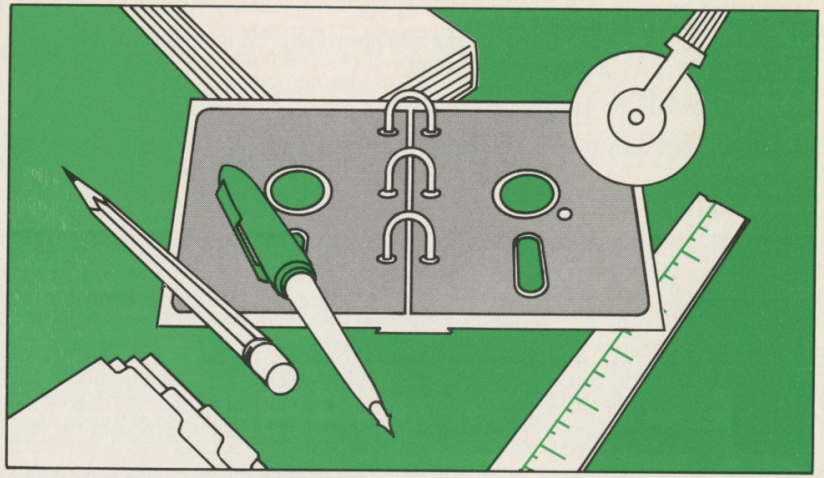
Following receipt of the Submission Agreement, the editors of *PC Disk* will evaluate the submission. At that point, we'll either indicate a lack of interest in the software or, if we are interested, we'll issue a submission authorization number and ask you to provide us with an executable copy of the program along with source code and documentation on 5¼" diskettes. Again, do not send us any additional materials until you receive a submission authorization number.

If we subsequently decide that the software is suitable for publication in *PC Disk*, we will offer a Software Contract, which will include such items as royalties, advances, and program and documentation changes required (if any). You will be asked to complete and test any program modifications agreed to in the Software Contract, and *PC Disk* will conduct a formal validation of the program and documentation.

We estimate that this process, from our initial evaluation to publication, takes approximately three to six months. This estimate is dependent upon a number of factors, and the process may take more or less time for your submission.

Our issues to date have already contained a number of programs received through our Software Submission Plan. The opportunity is real, so don't be shy. We look forward to hearing from you.

EDUCATION



BASIC BOOTSTRAPS 1

By Stan W. Merrill

Special Requirements: None
Files Used: BOOT1.BAS

The saying goes that you learn something new every day. In keeping with our own spirit of open idea exchange, we introduce BASIC BOOTSTRAPS, *a new feature meant to give you keener insight into the software driving your PC. By guiding you through a tutorial that explains the code behind a set of BASIC operations, we hope to give you a solid background so that you can begin to write your own BASIC programs. We have titled this feature* BASIC BOOTSTRAPS *because just as you start up your PC by "booting" the system,* BOOTSTRAPS *will "boot" your knowledge of the BASIC programming language.*

BOOTSTRAPS *will be a recurring feature, with each article exploring different aspects of BASIC. The program in this particular issue consists of six subroutines dealing with character graphics. As you delve into the code behind each subroutine, you will learn invaluable programming methods that you can use to make your own programs operate faster and do more.*

CHARACTER GRAPHICS

Character graphics are used primarily to generate text displays. There are three other graphics modes: medium-resolution, high-resolution, and user-definable. We will concentrate only on character graphics here and leave the other modes for future tutorials.

While the other graphics modes can be used to create sophisticated displays, character graphics does have some advantages:

—It is the only mode available with the monochrome monitor.

—It simplifies drawing forms, charts, boxes, and mazes.

—It is the only bona fide method of displaying more than four colors on the screen of a PC with a color monitor. Up to 16 colors are available.

START-UP

To start the *BOOTSTRAPS* tutorial, put a system disk in your default drive and load Advanced BASIC into your PC by typing:

BASICA [↵]

Insert the work copy of your *PC Disk Magazine* diskette in your default drive and type:

RUN "BOOT1 [↵]

The program begins by displaying the Main Menu, which describes each subroutine and the function keys used to activate each one. Included are the program line numbers at which the subroutines begin, so that you can pinpoint the location of each one.

After you run each subroutine of the *BOOTSTRAPS* tutorial, you will want to view the corresponding code so that you can follow our discussions of the programming techniques involved. There are two ways of doing this. You can either display the code directly on the screen or print out the entire program for easy reference.

To view the code on your monitor, return to the Main Menu and exit to BASIC by pressing:

[Esc]

Then use the LIST command to display specific line(s) of code. For example, display lines 60-140 by entering:

LIST 60–140 [↵]

If the code scrolls off the top of the screen too quickly, press:

[Ctrl] [Num Lock]

to freeze the display. Resume scrolling by pressing any key.

If you would like to have a hard copy of the program, return to the Main Menu and exit to BASIC by pressing:

[Esc]

Then make sure your printer is ready and type:

LLIST [↵]

When the listing is complete, restart the program by typing:

RUN [↵]

When listing the code, you will see that each program segment is delimited by header information. Included are the subroutine name, any variables that are created in this subroutine (listed after "Returns"), and the requirements (for example, variables created by other subroutines that must exist for this segment to run).

Once you have finished examining each section of code, explore the other features of *BOOTSTRAPS* by running the main program as previously described.

THE IBM PC CHARACTER SET

The first option on the Main Menu initiates the subroutine "Characters I have met". Press:

[F1]

to display the complete set of IBM PC characters, including the nonprinting control characters. Among the 256 available characters are alpha characters (A, B, C), numeric characters (1, 2, 3), punctuation marks (", ., :), and special characters (*, @, #). All of these can be used to create pictures on the screen. The highlighted number preceding each character is the number you would use in a program to represent that character (commonly called its ASCII code). Most of these symbols and their ASCII codes are found in Appendix G of the IBM BASIC Manual and Appendix C of the IBM Technical Manual. To examine the program code that creates the character set, exit to BASIC and type:

LIST 1460–1730 [↵]

The first major line of code is found in line 1540; these statements tell the program where the display memory is located in the computer by pointing it out with the BASIC DEF SEG statement. Lines 1570–1610 loop through all 256 characters, finding a memory location for each one. Every location is then colored red (which appears as normal on the monochrome monitor) and filled with a character.

Before writing the code to perform such an operation, you must first understand how the display memory corresponds to what you see on the screen. The screen is a rectangle of 25 rows and 80 columns of character cells. Each cell on the text screen is made up of a character and an attribute, one byte for each. An attribute is a notation that defines the nature of the character. In our example, the attribute is the color red, which causes the character to appear normal on the monochrome screen. Since each character cell requires two bytes of memory, think of memory as a rectangle that is 25 rows of 160 bytes. However, you cannot locate a memory byte by telling the computer the row and column you want. You must tell your PC the memory address (sequential numbers from 0 to 3999) that corresponds to that particular row and column location.

One programming method to use when setting up the text display is the BASIC POKE statement, which inserts the attribute and the character at the desired memory locations. In our example, line 1560 skips the memory locations of the first three lines of the screen that contain the title. Line 1580 sets the spacing for the character columns. Line 1590 calculates the address of the attribute byte, which is the second byte of each character pair, by spacing over 12 bytes and adding one. The POKE statement is then used to insert the numeric value for the color red, which was set earlier in the program. Line 1600 uses the same formula to calculate the byte just before the attribute byte. The subroutine POKEs the ASCII code for the character in memory and displays it on the screen. The subroutine looks around to the next character. Finally, lines 1660-1720 position the cursor at the proper row and column and print the character's ASCII code.

BOXES

One of the features of the character graphics mode allows you to create boxes and forms. To view some of the shapes that can be drawn, press:

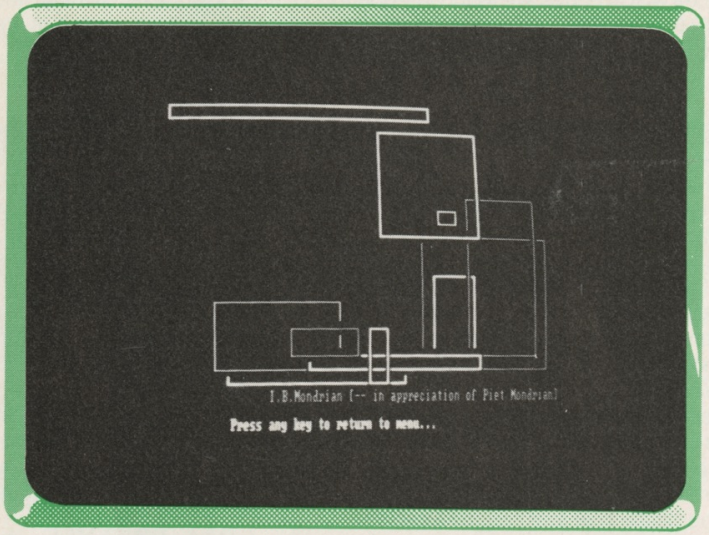
[F2]

from the Main Menu. The subroutine draws boxes with random sizes, colors, and locations. To view the program code, exit to BASIC and type:

```
LIST 1740-2020 ↵
```

You can also use this subroutine to demonstrate how to draw forms in general.

Before beginning to write code for shapes and forms, you must decide which graphics characters to use. The character set in this tutorial should be sufficient but if it is not, refer to Appendix G of the IBM BASIC Manual for additional characters. The boxes in our example have four corners (characters 218, 191, 192, 217), a horizontal line (character 196), and a vertical line (character 179). Lines 1790-1810 identify these characters as "constants". These are single-line boxes,



Design exotic artwork

but you can make double-line boxes by inserting the appropriate character numbers. Finally, lines 1820-2010 position the cursor at the corners of the boxes and connect them with vertical and horizontal lines.

This subroutine requires additional input before you can execute it. You must specify the exact location and size of your forms in a separate subroutine, an example of which is shown in lines 2030-2280. Here the values for the row numbers for the top and bottom locations of the boxes are specified with the TOP.OF.BOX and BOTTOM.OF.BOX statements. The column numbers for the left and right sides of the boxes are specified with the LEFT.OF.BOX and RIGHT.OF.BOX statements.

There is another character-printing method that you can incorporate in your programs. In BASIC, C, Pascal, Fortran or any other programming language, you can print a character by referring to its ASCII code. In BASIC, the function CHR$() is used with the PRINT statement to do this. For example, the statement PRINT CHR$(65) displays an "A", PRINT CHR$(66) displays a "B", and so on. To print the double horizontal line in our example, you could put the statement:

```
PRINT CHR$(205);
```

in a program loop.

GRAPH PAPER

When working in the character graphics mode, you must frequently determine exactly what belongs in every one of the 2000 (80 X 25) character cells on the screen. A good way to do this is to sketch your picture on graph paper first. Unfortunately, most graph paper has square or logarithmic cells, while character cells on the screen are rectangular. To solve this problem, use the graph paper function found in lines 2330-2530. This subroutine will print graph paper suitable for laying out screen designs. To run this subroutine, press:

F3

from the Main Menu. You will be prompted to make sure your printer is ready. There are no error handling routines in this program, so if your printer is turned off, your system will hang. (Error handling will be discussed in future *BOOTSTRAPS* columns.) Next press any key to begin the printout. When you have printed enough graph paper, return to the Main Menu by pressing:

Esc

and then the Space Bar. This routine prints graph paper using the number sign (#) and the plus sign (+). If you wish to use other characters, just change the MARK1$ and MARK2$ constants in line number 2340.

MUG SHOT

This segment of the tutorial shows you how to use the graph paper you just printed to generate figures on the screen. First, execute the program to see an example of what you can create. From the Main Menu, press:

F4

and the image of a mug will be displayed.

The first step in creating such an image is to draw or trace it on your graph paper. Any character inside the outline of your image is to be included as part of the object and anything outside is to be the background. Next, choose which characters and attributes you want to represent the background and any highlighted or shaded areas of your image. The characters and attributes used in this subroutine to draw the mug have been input as DATA statements in lines 2720-2930.

The logic behind this subroutine is that contiguous cells on the screen are often similar in attribute and character. With the mug, for instance, much of the screen is background composed of characters in the color white. The subroutine reads the contents of the DATA statements and determines how many characters are contiguous, what the number of the character is, and what the associated attribute should be. The program then draws the appropriate number of the specified character and proceeds to the next group of contiguous characters.

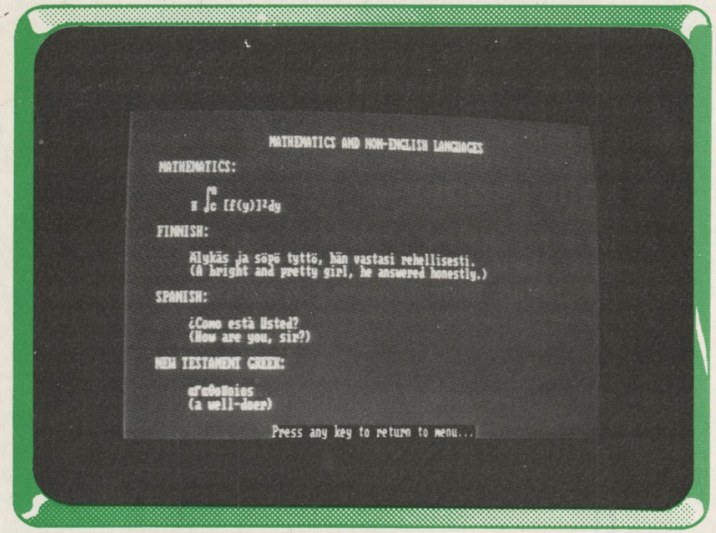
COLOR SWATCHES

To appreciate the material in this section of the *BOOTSTRAPS* program, you must use a color monitor. Most of the screen displays in *BOOTSTRAPS* are done with various foreground colors on a black background. Since character graphics is the only graphics mode with more than four colors (up to 16 are available), you can use different colors for the background as well as the foreground of a character cell. If you do this with characters that fill the entire character cell, you can create different combinations of shading and thus obtain a wider range of colors for your graphics displays. However, the screen resolution in this mode is low, which prevents you from drawing smooth curves or fine details.

Choose option 5 on the Main Menu to run the Color Swatches subroutine. The re-

sulting screen displays the available background and foreground colors. The code that creates this display begins at line 2950 of the main program. This subroutine consists of a series of nested FOR-NEXT loops that cycle through the possible colors and display them using the foreground and background of the character with ASCII code 177. The PC can use up to four characters to generate the shading effects shown in the program. The ASCII codes for these characters are 176, 177, 178, and 219. We have chosen character 177 here because it has the same number of foreground and background pixels. This way we can mix a red background with a white foreground and get pink, mix red with yellow to get orange, and so on.

MATHEMATICS AND NON-ENGLISH



The character combinations are endless!

Character graphics can also be used for text processing in non-English (primarily Western European) languages, and for writing mathematical formulas. Press:

F6

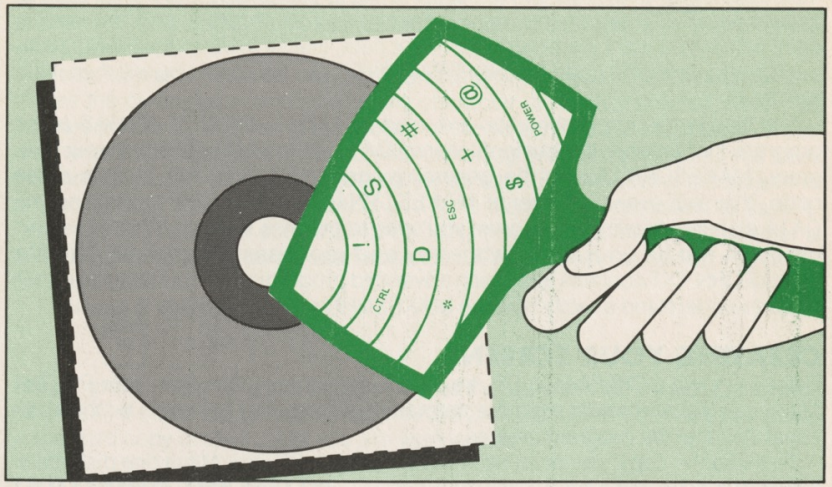
to run the Mathematics and Non-English subroutine. This subroutine displays some of the languages and mathematical symbols that you can create.

This section of code consists of a series of PRINT statements that display the variety of symbols and characters that you can display on your PC. To generate these characters yourself, press:

Alt

and simultaneously enter, via the numeric keypad, the number of the character to be displayed. You can use the existing characters or combine them to create your own symbols. To see the available characters and their associated ASCII codes, consult Appendix G of the IBM BASIC Manual or run the "Characters I have met" section of this tutorial.

UTILITIES



SCREENMAKER

By Gary Oppenheimer

Special Requirements: None
Files Used: MAKESCRN.BAS
MAKESCRN.USE
MAKESCRN.TXT

Programming on the company's new PC was going to be so enjoyable and productive. You were going to stay late at the office for a week or two, study the BASIC manuals, and develop programs perfectly tailored to the firm's needs.

Only it hasn't quite worked out that way. In your BASIC studies you've gotten only as far as PRINT and GOTO and IF . . . THEN . . . ELSE, and it looks like it's going to be a long haul to the money- and time-saving application programs you dreamed of (and promised your boss).

Don't fear for your job or for your self-esteem! Don't call an outside programmer! Consult PC Disk Magazine *by running* SCREENMAKER . *This program creates powerful data-entry code, custom-fitted to your specific application. Once* SCREENMAKER *has done the hard programming, all you have to do is write a simple routine to process the data in any way you wish.*

BACKGROUND

Data entry can be one of the most sophisticated and tedious types of programming. Prompts must be given, entries must be checked for all sorts of errors, and the screen must be formatted so that it is attractive and understandable to the program user. *SCREENMAKER* allows you to create program code that will perform these complicated data-input functions. You can then incorporate this code into your own data processing program.

SCREENMAKER helps you create a data-input screen for your program. You can arrange the fields (the areas where data is entered) in any configuration you wish, specify what type of information is to be entered in each field, and dictate what sort of checks should be performed on the input.

SCREENMAKER works by creating DATA statements for you. You incorporate these DATA statements into your program code along with the code found in the MAKESCRN.USE file. This file contains the READ statements that will be used to access your DATA statements. As you are writing your program, and you reach a point where you want the program to input data, just GOSUB to the READ statement subroutine. The code that *SCREENMAKER* has made for you will then input the data.

The files in this implementation are used as follows: MAKESCRN.BAS is the program that creates the custom-tailored DATA statement section of your program. MAKESCRN.USE is the general program that uses this custom-made code. You will eventually merge MAKESCRN.USE and the file containing the DATA statements with the program you write to process the data. This three-part program will be your final data entry and processing program. The file MAKESCRN.TXT contains text which advanced programmers may wish to read in order to understand exactly how the MAKESCRN program works.

CREATING YOUR SCREEN

Before running *SCREENMAKER* , transfer the necessary files from your *PC Disk Magazine* diskette onto a separate, formatted diskette that has room for the additional files that the program creates.

To make the copy, you must be in DOS. Put your *PC Disk Magazine* diskette in the default drive and type:

```
COPY MAKESCRN.BAS B: ↵
```

where "B" is your destination drive (on a single-drive system, DOS will prompt you to swap in your destination work diskette for each COPY command). Repeat this procedure for the additional files to be copied, MAKESCRN.USE and MAKESCRN.TXT.

To begin creating a custom-made data processing program, load Advanced BASIC into your PC by putting a system disk in your default drive and typing:

```
BASICA ↵
```

Then put your work copy of *SCREENMAKER* into the default drive and type:

```
RUN "MAKESCRN ↵
```

SCREENMAKER displays a Work Screen with a row of function key prompts at the bottom. First press:

F4

to clear the screen of title information. (This key will clear the screen at any time.) The cursor then moves to the upper left-hand corner of the screen and awaits input. You are now ready to format your input screen. Use the cursor control (arrow) keys and the Home and End keys to move the cursor to the position on the Work Screen where you want the first input field to appear. The Home key returns the cursor to the upper left-hand corner of the Work Screen, while the End key moves it to the lower right-hand corner. Type in:

```
1.
```


This tells the program that this is field number 1. Now you will probably want to type in a prompt, such as "Name:", "Amount:", or "Date of last payment:". Follow this with a string of underline characters (—). These characters alert the program to the actual location of the entry field. Be sure you type enough underline characters to accommodate the data to be entered in this field. If you expect long names, leave plenty of room, but if you know the input will be a seven-digit telephone number, you need only include eight underline characters (seven numbers with a hyphen).

Create as many fields as you can fit on the Work Screen. The Backspace key moves the cursor back and erases any character previously entered. Note that the cursor location is given at the bottom of the screen by row and column. If you wish to stop in the middle of the screen-formatting process, the F1 key will save the current screen so that it can be called up later. Before saving the screen in progress, *SCREENMAKER* prompts you for a filename under which to save it. There is a default name of MAKESCRN.SCN. You can use this name or enter a different one, but it is a good idea to use the .SCN extension to remind you of the file's contents.

To reload a screen, press:

F2

The program asks you for the name of the file that contains the screen to be loaded. Again, it offers the default name, MAKESCRN.SCN. The F6 (Print) key gives you a hard copy of the screen you are working on.

TAILORING THE FIELDS

Once you have arranged the screen to your satisfaction, press:

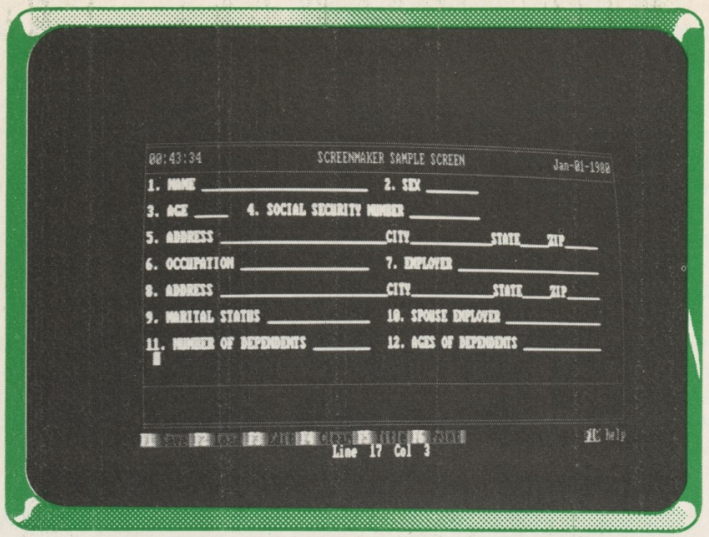
F3

This Edit function allows you to specify many helpful parameters for the data input. First, choose a name for the file in which *SCREENMAKER* will store the DATA statements that it is creating. The program offers a default name of SCREEN.DAT, and again, it is a good idea to use at least the extension .DAT to remind you that this file contains DATA statements.

Next you are prompted for a line number at which these DATA statements should begin. The program suggests 60000 as a default value. This means that the program you create around *SCREENMAKER* 's code will not be able to use these line numbers. (*SCREENMAKER* uses lines 40000-60000 for its code). Select a line number and proceed.

The program pauses a moment to process the screen and then prompts you to make certain choices about how to handle the input for each field. Field number 1 will be highlighted on the screen, and at the bottom of the monitor, the program will ask what type of information will be entered in that field. The choices are Text (names, addresses, titles, categories), Integers (any sort of number), Dollar (a number with no more than one decimal point and two decimal places), Date (entered in the form MMDDYY—6 characters only, with no slashes or spaces), Function Key, and Display Only (for data that should be typed in for reference but not processed). The program prompts you to "Enter a number, not a blank. Press Enter for Default." The cursor is now at the bottom right-hand corner of the display, with the numeral "1" already entered as the default. If you are not sure what you are doing here, press the F10 key to display a help message.

SCREENMAKER asks several other questions about each data field. First, you are prompted to choose whether you wish to convert your input to upper case. Answer "yes" if you want your program to convert to upper case any lower case characters typed into this text field. Next, you can elect to erase the field after your



Design your own data-input screen

input is complete. This option causes the program, upon execution, to erase this field from the data-entry screen once the data has been entered into it. The data would remain in the field, but neither the data nor the field prompts would be displayed while the user inputs other data fields.

If you wish to limit your input to a range of values, answer "yes" to the appropriate prompt. Then enter a starting value (default value 0) and an ending value (default value also 0). The next prompt, "Entry requirement: 1-Full, 2-Partial, 3-Optional"—allows you to define whether the user must make an entry in this field and further, whether the field must be filled completely upon entry. Finally, an option allows you to restrict a field's input to certain multiple choices, such as "Y" or "N", or "Male" or "Female". If you choose this option, *SCREENMAKER* next prompts you to input these choices on one line, with a slash (/) after each (including the last). Later, when a user tries to enter data into this field, the program that *SCREENMAKER* creates will tell him his choices, and give him an error message if he tries to enter anything else.

The program prompts you for a Help message for each field. This can be up to 65 characters long, and is displayed whenever the user presses the F10 key during data input.

SCREENMAKER also allows you to set a default value for each field. This value appears in the field on the screen while the user is entering data for that field. The user can enter the default value merely by pressing the Enter key.

Once you have answered all the appropriate questions for each field, *SCREENMAKER* creates a file (using the name you specified) that contains DATA statements. You can now exit the *SCREENMAKER* program by pressing:

Esc

Doing so returns you to BASIC.

If you wish, you may look at the DATA statements that *SCREENMAKER* has created by returning to DOS and using the TYPE command. Simply enter:

"TYPE filename"

at the DOS prompt and the DATA statements will be displayed. The text in the file

MAKESCRN.TXT explains in detail how the *SCREENMAKER* program has coded all your information into these DATA statements. As you become more familiar with *SCREENMAKER*, you may wish to use this information to edit the DATA statements directly (using a standard text editor), thus bypassing the screen-formatting part of *SCREENMAKER*.

MERGE AHEAD

You now have two of the three components of your final applications program: the file containing the DATA statements you just created, and the MAKESCRN.USE file containing the program code that allows your program to read these statements. The next step is to merge these files with the BASIC MERGE command. From Advanced BASIC, first clear your PC's memory by typing:

NEW ↵

then load the MAKESCRN.USE file into program memory by typing:

LOAD "MAKESCRN.USE" ↵

Then load the program that contains the DATA statements into memory as well by typing:

MERGE "yourfile" ↵

Where "yourfile" is the name of the DATA statements you created with *SCREENMAKER*. You now have the data-input part of your program in memory. Now write this file to disk for safety by using the BASIC SAVE command.

WRITING YOUR PROGRAM

You are now ready to write the data processing part of the program. When doing this, you must be aware of several facts concerning the sections of the program that *SCREENMAKER* has created for you:

—They occupy lines 40000 through 59950, plus the lines you chose for the DATA statements. The code that you create to process the data should not use these program lines.

—They use only variables prefixed with "S." (S.INPUT%, S.DA$, S.LEN%, etc.), so you must not use any variables with the "S." prefix.

—They must be initialized before they can be used. This is done with a subroutine contained in the code at line 59000, so one of the first lines of your program should contain the command GOSUB 59000.

The data-inputting program is contained in a subroutine that begins at line 40000. But before you GOSUB 40000 you must specify which field in the screen you wish to obtain data for. Simply set the variable FI% to the field number.

GOSUB 40000 causes the program, upon execution, to display the screen that you created. The field that you specified with the variable FI% will be flashing, waiting for input. Whatever input the user types in will appear on the screen in the field. If the user needs help, he or she can just press the F10 key to view the help message you created for that field. The program will also prompt the user for the right type of input, and give an error message if the wrong type of input is entered. Once the user has pressed the Enter key, the input will be stored and the program will return from the subroutine at 40000.

The data that was entered is stored in the variable S.INPUT$. It is also stored at a location in the array S.DA$(FI%), where "FI%" is the number of the field. S.INPUT$ is updated with the latest input data each time the program goes to the line 40000 subroutine. But the information from the previous field will still exist in the array, so it is possible to collect the data from an entire screen before having to process it. The easiest way to do this would be with a FOR . . . NEXT loop.

SCREENMAKER makes this process easier with the variable S.I0%, which always contains the number of fields in the screen. A standard application of the routines created by *SCREENMAKER* might look like this:

```
100 GOSUB 59000
200 FOR FI% = 1 TO S.I0%
300 GOSUB 40000
400 NEXT
```

This routine would initialize the variables, set FI% to 1, perform the data-input routine on field number 1, and store the entered data in S.INPUT$ and S.DA$(1). It would perform the data-entry routine again for field number 2 and store the data in S.INPUT$ (erasing the data from field number 1) and S.DA$(2). This procedure would be repeated as many times as there are fields (S.I0% times) and then proceed to the program lines after 400.

These lines ought to contain code to process the data that has been collected. For example, if each screen were used to input weekly earnings data for an employee, this code might multiply work hours times hourly rate, put the result in an array of variables which holds weekly pay for each employee, and add the result to a running variable which contains total labor costs for the week.

SPECIAL FEATURES

The program code that *SCREENMAKER* creates for you has a couple of helpful features. The first is a variable called S.EXIT%. This variable is initialized to 0. However, if during the execution of the data-input portion of your program the user presses the Escape key, the *SCREENMAKER* code changes this variable to a nonzero number. Therefore, if you check the variable S.EXIT% before the NEXT statement in the FOR . . . NEXT loop, and go to an exit routine if it is not equal to zero, you could instruct your user to press the Escape key in order to exit the data-entry program.

It is also good to check the variable S.INPUT% after the GOSUB 40000 command in each loop. You would do this if you expected the user to press a function key when entering data in a field. If a function key is used as entered data, the program will not treat it as normal data. It will set S.INPUT% to an integer, 1 through 9, depending on which key was pressed. But it will not store the number in the S.DA$ array. Therefore, the function key option should be used in a field not to enter data, but to give the user some options in program operation.

For example, if you were using the screen to input detailed scientific data, you might set a field to accept function keys F1 or F2 as input and instruct the user to press F1 if the information is to be rounded off or to press F2 if it is to be kept to a precision of, say, three decimal places. You would then put some checking steps in your program before the NEXT statement in each loop. You would first see if FI% had reached 5 yet. If it had, you would check to see if S.INPUT% were 1 or 2. If it were 1, you would send the program to a rounding-off subroutine. If S.INPUT% were 2, you would send the program to a three-decimal-place precision routine.

Another helpful variable is the array S.LEN%(FI%), which contains the length of each field (the number of underline characters you initially typed in). Note that while the fields used for entering dates need only 6 places, you must leave extra room when formatting your screen. When the user inputs a date in the form MMDDYY, the inputting program redisplays it as MM/DD/YY.

Note that the *SCREENMAKER* code stores all variables as strings. If the variable is numeric, you will probably have to convert it with the BASIC VAL command before you can process it. Note also that if input data does not fill the entire length of the field, *SCREENMAKER* will store it either right-justified (if it is numeric) or left-justified (if it is non-numeric).

PROBLEM RECOVERY

There really is not a great deal to say about problem handling with *PC Disk*. If you use this software on the right equipment running the appropriate system software, as outlined in the Technical Preface, you should experience no problems. Nevertheless, a few comments may resolve some more obvious difficulties.

Any BASIC program can be interrupted at any time by pressing:

If you do not see the Ok message immediately, indicating that you are back in BASIC, press these keys again. This is a rather drastic but effective way of regaining control of the computer. You won't damage any of the programs in this way, since they're still intact on the diskette. However, you may lose data you entered while the program was running.

If you interrupt a program, you may find that the function keys no longer perform as they had before starting the program. This is because many *PC Disk* programs reset the function keys during execution, then restore the original settings upon completion. An interrupt causes an abnormal termination of a program, so the function keys are not restored. To correct this situation, simply exit from BASIC and then return to BASIC.

You may find at times that the cursor control keys are not working as they should. This is because the keys are not in cursor control mode. The key that switches these keys between numeric mode and cursor control mode is the Num Lock key. So to restore the keys to cursor control mode, press:

If you try to send something to the printer when there is no printer, or when the printer is off or offline, you can hang the system. The computer will just sit there and will not respond to any keys pressed. After a few seconds you may get a BASIC error message indicating that the device was unavailable. The program that was running has been aborted, and you will be left in BASIC. If the computer does not put out any message, but just remains hung, you will have to say good-bye to whatever you were doing and re-boot your system.

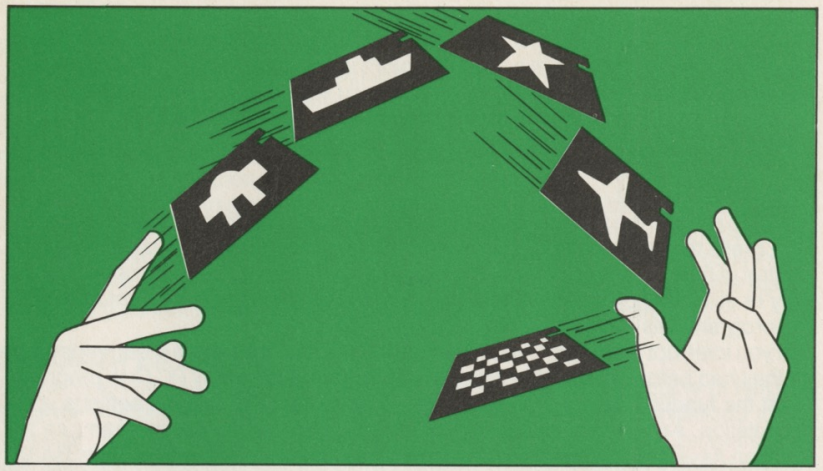
Though we hope it will never happen, if you should find a "bug" in a *PC Disk* program, *please do not send back your diskette*. Please call:

(212) 725-7773

or write:

PC Disk
Problem Handling
One Park Avenue
New York, N.Y. 10016

GAMES



SECRET SEQUENCE

By Ralph G. Brickner

Special Requirements: None
Files Used: SECRET.EXE

Chances are you've played games against the computer before, but has the computer played back? In SECRET SEQUENCE, *the computer is both player and challenger. This game requires cunning and logic as you try to guess numerical sequences generated by the program and the computer tries to guess sequences you create.*

PROGRAM STRUCTURE

In *SECRET SEQUENCE*, play alternates between you and the computer. At the beginning of the game, the computer generates a sequence of four digits. You have ten chances to guess this sequence. Whether or not you are able to discover the sequence in ten tries, it then becomes your turn to make up a secret sequence. As the computer makes its own numerical guesses, you will tell the program how many of the digits in the computer's guess are correct and how many are correct but are misplaced in the sequence.

STRATEGY

To play *SECRET SEQUENCE* successfully, you must develop deductive skills. Although you might have to play a hunch for your first guess, you must use logic on subsequent tries. With the computer's clues as a guide, try to narrow down the possible numerical combinations in order to arrive quickly at the solution. Every time you make a wrong guess, the computer receives a point; later, you will earn points every time the computer errs when guessing your sequence.

START-UP

Insert your work copy of the *PC Disk Magazine* diskette in your default drive and type:

SECRET ↵

After viewing the Title Screen, press any key to continue. Next, the Help Screen briefly explains the game.

You must set the level of difficulty for each game by selecting, at the outset, the range of digits to be used. Each "secret sequence" will consist of four digits, but you can limit the range of possible choices. The easiest level permits you and the computer to use only the digits 1-4. The most difficult level of play allows eight. (These numbers are 1-8; 0 and 9 are never used in *SECRET SEQUENCE*.) Obviously, the first level is easier because it allows fewer possible four-digit combinations than eight different digits would. There are 256 possible arrangements for a four-place sequence using the numbers 1-4, but 4096 possible permutations when each digit is a number from 1-8.

At the bottom of the Help Screen is a prompt for you to enter the number of different digits you wish to use. There are five possible categories to choose from: 1-4, 1-5, 1-6, 1-7, or 1-8. Enter one of these choices by pressing a number from 4 to 8. If for example you enter a 5, this round of the game will be played using only digits 1-5. Thus, for this particular numerical sequence, the first digit would be any number from 1-5, as would the second, third, and fourth digits. It is possible to repeat any number two or more times in a sequence. Possible sequences would include 1245, 3355, 5413, and 5555; 6430 would not be permissible because 0 is never used and because 6 is beyond the range of digits specified for this round.

Note that it is not possible to change the level of difficulty in the middle of a game. If you originally opt for level 4 and wish to advance to 8, you must start the game from the beginning. Make your choice carefully; then press any key to go to the next screen. At the top of the display is the title, "Secret Sequence—Human Guessing." You will start the game in the role of "Seeker of the Sequence".

SEEKER OF THE SEQUENCE

Whenever the game begins, the computer is the first Keeper of the Sequence. Accordingly, your electronic opponent will select the first sequence, using any four-place combination of those digits you specified in the Start-Up procedure.

As the Seeker, you must guess the computer's secret sequence. Begin by entering four digits, one at a time, in response to the program's prompt. After you type each digit, the flashing question mark which serves as the cursor will move into position for the next digit. After you have typed four digits, the program asks you to verify your choices. If you are satisfied with your guess, press:

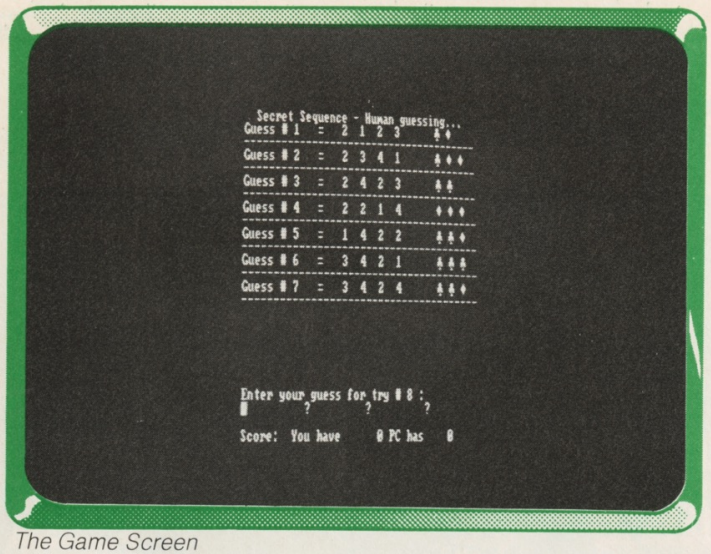
Y

Otherwise, type:

N

and retype four digits.

Once you confirm your sequence, it appears at the top of the screen for evaluation. The Keeper uses a "club" to mark each digit that is correct and in the right place in the sequence. A "diamond" indicates each digit that appears in the computer's sequence but is misplaced in your guess. For example, assume that the



The Game Screen

computer's secret sequence is 2214. If your first guess is 1234, 2 clubs and 1 diamond will appear. The clubs represent the 2 and 4, both of which are correct and in the right place. The diamond represents the 1, which is correct but in the wrong place. Nothing appears for the 3, since it is entirely wrong. Keep in mind that the locations of the clubs and the diamonds on the screen do not refer to specific digits in the sequence. The number of clubs shows how many of your digits are completely correct; the diamonds simply indicate how many are right but in the wrong place. It's up to you to figure out which digits are correct, and which need to be rearranged.

Examine the clues before entering subsequent guesses. Your previous guesses and the computer's clues will remain on the screen until your turn is over so that you can use them for comparative reference.

You have ten chances to figure out the sequence generated by the program. The computer receives one point for every wrong sequence you enter, and one additional point if you do not succeed in ten tries. Once you have determined the correct sequence, it is your turn to make up a secret sequence. If you do not succeed after ten guesses, the computer awards itself 11 points, and it is then your turn to play as "Keeper of the Sequence."

KEEPER OF THE SEQUENCE

Now create a sequence of your own and challenge the computer. Once you have a sequence that complies with your choice of digits in the Start-Up procedure, write it down so that you can refer to it during the computer's turn. The computer's guesses will appear at the top of the screen. At the bottom, a prompt will ask you first to indicate how many of the computer's digits are correct and in the right place. For example, if only one digit in the guess is correct *and* in the right place, you would type:

1

Next the program asks you to indicate how many of the computer's digits are correct but in the wrong place. If for example three of the computer's digits appear in your sequence but in different locations from those the computer indicated, you would answer:

3

After you type in this number, you must verify that this information is correct. Check your secret sequence to make sure that you gave the computer the correct clues, then type:

Y

If you instead realize that you have made a mistake, type:

N

and you can reenter the information.

After verifying your information, the program displays clubs and/or diamonds beside the computer's guess based on the answers you supplied. When the PC is the Keeper, it will not give you false clues. However, if *you* err in providing clues for the PC, it may detect your error. This will happen as soon as the computer detects a logical inconsistency in your responses. If it does, it will respond, "I see you're confused . . . I'm afraid we'll have to replay this round. Press any key to continue . . . " If you choose to continue, a new game will start with you as the Keeper and the PC as the Seeker. It also awards itself three points for such mistakes, so double-check your clues.

SCORING

As game play continues, you alternate with the PC in the Seeker and Keeper roles. The score is displayed at the bottom of the screen. Under normal circumstances, only the Keeper earns points during a round, up to a total of 11 points if the Seeker fails to guess correctly on all ten tries (remember that in this case, the Keeper gets a bonus point). You cannot get points when you are the Seeker, but the PC can earn three points each time you supply the computer with erroneous information when it is trying to guess your sequence.

EXITING

There are two ways to end *SECRET SEQUENCE*. At the end of every round, the question, "Replay (Y/N)?" appears. To stop the game at this point, type:

N

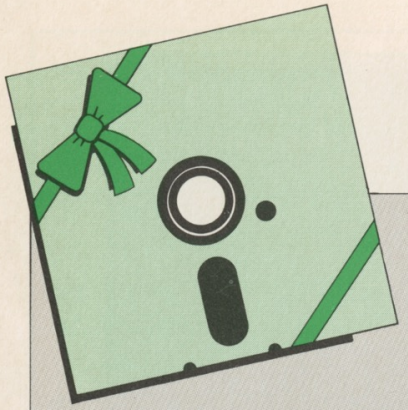
Otherwise, you can exit the game at any time by pressing:

F1

SUBSCRIPTION INQUIRIES

If you have any questions regarding subscriptions, or you would like to renew your subscription or order back issues, call:

800-932-0017
in N.J. 800-624-0497
or write:
PC Disk
P.O. Box 5930
Cherry Hill, NJ 08034



EXPAND YOUR SOFTWARE LIBRARY

This issue of *PC Disk* has already increased your software library for your IBM Personal Computer. If you haven't subscribed to *PC Disk*, just think of how easy it will be to expand your library with future issues delivered to your door at substantial savings. And if you already are a subscriber, wouldn't you like to share your knowledge with friends and get them gift subscriptions to *PC Disk?*

Either way, you can save $60 with this introductory offer and give yourself or a friend up to 40 programs for less than $4.00 each! *PC Disk* is the solution to the problem of building and maintaining an up-to-date library of tested software programs.

You will get up to 8 varied and ready-to-use programs and files in every issue—complete with an illustrated user's manual—for the ultimate in ease, efficiency, and economy.

Every program in *PC Disk* is developed by experts, tested by the editors of *PC Disk*, and published by the people who bring you *PC*, guaranteeing quality and satisfaction.

Use the enclosed card to subscribe today or to give a friend a gift. If the card is missing, please write to:
PC Disk, PO Box 5930, Cherry Hill, NJ 08034
For even faster service, call our number below and use your American Express, MasterCard, or Visa . . . today!

CALL
800-932-0017
in N.J. 800-624-0497
and get 6 issues of
PC Disk
for only $149
and save $60!

COMING UP

STATISTICS LIBRARY 2
Continue to expand your library of statistical routines with the second installment of our statistics series.

STOCK MANAGER
A powerful system for analyzing equity investments. Use the market model to examine overall trends, while employing a number of technical analyses to evaluate individual stocks. Our first in a series of investment analysis and management programs.

NOTE MINDER
Organize those unwieldy notes and reminders into one central file. Search on a key word to find the note you want quickly and easily.

BASIC SHRINKER
Eliminate the wasted space in your BASIC files. Shrink your files to make the most out of every sector on your disks.

TIC TAC GO
Test and improve your perceptual skills as you try to match patterns in this unique and original game.

FLASHKARD
An educational tool that uses your PC to present a tried-and-true learning technique, flashcards. Create cards for any subject you like with the program's flexible format.

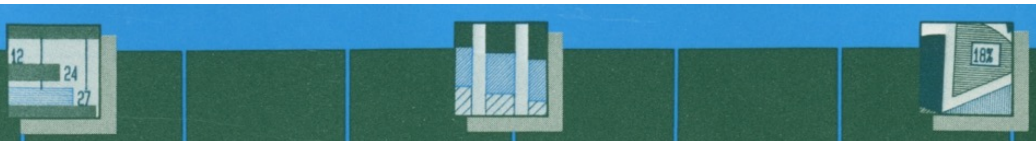
BACK ISSUES OF ..... PC DISK

Add to your *PC Disk* collection today, while copies are still available. The more complete your library of back issues, the more enjoyment and use you can get from your personal computer.
The currently available back issues begin with Volume 1, Number 1. Order any issue you want, being sure to specify volume and issue number, and quantity, for each. This offer is subject to availability. If a particular issue is out of stock, your payment will be refunded promptly.
Back issues of *PC Disk* are priced at $31.95 each, which includes postage and handling. Please write to:

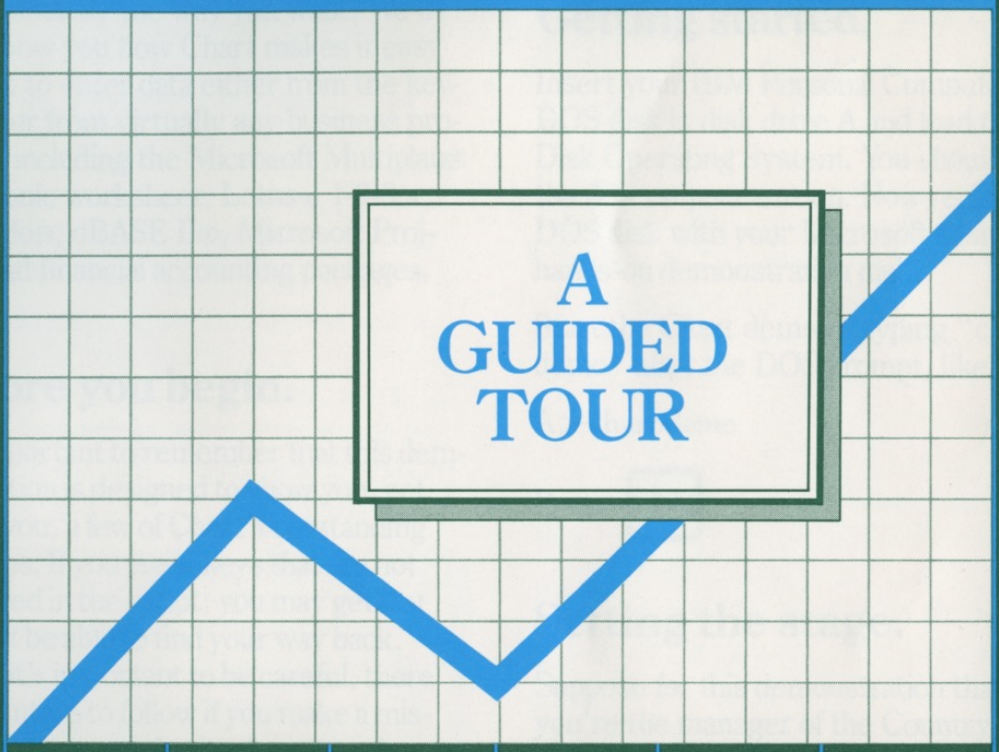
PC Disk
P.O. Box 5930
Cherry Hill, NJ 08034

Be sure to include volume and issue number, as well as the number of copies desired, and the address to which you would like those copies sent. Please enclose your payment with your order. For faster service, call:

800-932-0017
in N.J. 800-624-0497



MICROSOFT® CHART



A GUIDED TOUR

SPECIAL DISK INCLUDED

Welcome to a hands-on demonstration of the business graphics program that helps you communicate data clearly and persuasively.

We'll show you how Microsoft® Chart lets you choose from a large selection of pre-designed formats; and how Chart gives you the flexibility to design custom charts exactly the way you want. We'll also show you how Chart makes it easy for you to enter data either from the keyboard or from virtually any business program, including the Microsoft Multiplan® electronic worksheet, Lotus™ 1-2-3™, VisiCalc®, dBASE II®, Microsoft Project, and financial accounting packages.

Before you begin.

It's important to remember that this demonstration is designed to show you, not teach you, a few of Chart's outstanding features. If you press keys that are not indicated in the script, you may get lost and not be able to find your way back. While it's important to be careful, there are pointers to follow if you make a mistake. For a more thorough explanation on using Microsoft Chart, we suggest that you refer to your local computer store dealer.

Remember to watch your screen while you're going through the demonstration. It's easy to become absorbed in what keystrokes to press, and forget to watch Chart in action.

For this demonstration you'll be using a special copy of Microsoft Chart. While it's very similar to the actual product, you won't be able to save or print charts except for those indicated in the script.

System requirements.

You'll need an IBM® Personal Computer with at least 128K of memory (256K is recommended), one double-sided disk drive, and an IBM Color/Graphics Adapter or a Hercules™ Graphics Card.

Getting started.

Insert your IBM Personal Computer DOS disk in disk drive A and load the Disk Operating System. You should see the A> on your screen. Now replace the DOS disk with your Microsoft Chart hands-on demonstration disk.

Start the Chart demo by typing "chart demo" after the DOS prompt, like this:

A>**chart demo**

Press 

Setting the stage.

Suppose for this demonstration that you're the manager of the Country Shop, a local branch of a retail clothing store chain. You're concerned about the City Shop, a nearby competitor. It seems that the City Shop is doing better than you had expected, and you'd like to understand how it's affecting your business. Once you've analyzed this problem, you plan to present the results to your corporate office.

We'll show you how Microsoft Chart helps you communicate your data effectively—how you can get your point across *exactly* the way you want to present it.

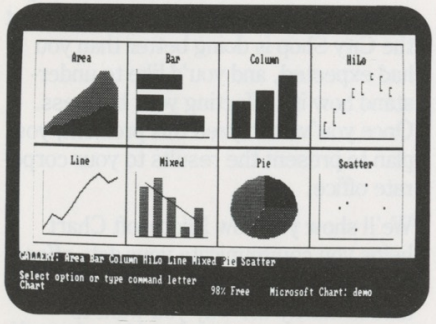
Choosing a chart from the gallery.

Let's start by looking at the simple pie chart on your screen. As you can see, this chart shows the Country Shop's retail customer traffic from July to December. December is highlighted, or exploded away from the rest of the chart, to call attention to what was a particularly good month for your store.

But a pie chart probably isn't the best way to illustrate customer traffic. It's hard to see the month-to-month differences. Pie charts are good for showing the composition of a whole—or pieces of the pie—but line, column, and area charts are better for showing trends or changes over time. Let's change this to an area chart.

Changing from one chart format to another can take a lot of time with other programs. But with Microsoft Chart you simply choose the chart that best suits your needs with *three easy keystrokes!* No changing disks. No reentering data. Sound impressive? It is.

Press G (Gallery)



That's keystroke 1. What you see on your screen is Chart's gallery of eight basic chart types. Area. Bar. Column. HiLo. Line. Mixed. Pie. And Scatter. Just pick the type of chart you need. There's no guesswork involved. We decided on an area chart.

Press A (Area)

That was keystroke 2. As you can see, there's more to Chart's gallery than first meets the eye. You now have a full selection of area charts to choose from. Ready for keystroke 3? Let's choose area chart 1.

Press ! 1

Chart automatically builds the chart for you. We could have built this chart using other Chart commands that are displayed at the bottom of your screen. But Chart's gallery let us build it especially quickly. And easily. It recentered the title, rescaled the axis, and refilled the screen. Automatically.

Looking at the area chart, you can see that it shows trends over time while also highlighting the absolute magnitude of your customer traffic. But what you really want to emphasize are the month-to-month differences in traffic. Chart offers 45 pre-designed formats to choose from. Let's go back to the gallery and try another.

Press G (Gallery)

A column chart is a good selection since it stresses more of a contrast between months.

Press C (Column)

Chart gives you eight different column charts to choose from. Column chart 6 shows the columns on a grid background, which will make your month-to-month differences come out even stronger.

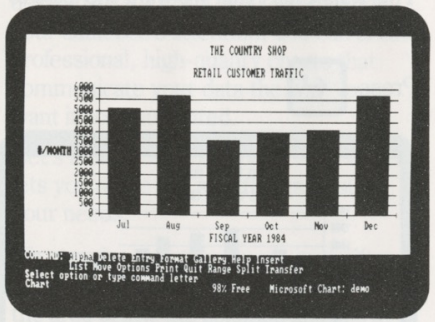
Press ^ 6

Now you have a chart that really shows what we're after. That's what's nice about the gallery. It's a built-in short-cut that gives you *immediate* access to 45 chart variations. You'll find it especially helpful if you're not sure what type of chart is best.

But Chart's gallery is only the beginning. Let's look at how we can tailor this gallery-built chart to one that meets your precise needs for the Country Shop.

Customizing your chart.

Chart's gallery has a lot to choose from, but there will be times when you'll want custom charts beyond the 45 available in the gallery. Microsoft Chart lets you customize your chart in thousands of different ways. Let's look at an example.



The column chart on your screen shows the retail traffic just for your store. To compare it with the competition, we'll need to add the retail customer traffic of the City Shop. First let's look at the list of data series that make up this chart.

Press L (List)

The list screen shows your data series. A "data series" is a number of related data points. The Country Shop and the City Shop are the two data series now showing on your screen. Both summarize customer traffic from July through December.

Notice that there's an asterisk next to the Country Shop in the "Incl.(*) Name" column, which means that the Country Shop is included in your current chart. To add the City Shop data, all you have to do is add an asterisk next to the City Shop. This tells Chart that both series are to be included in the next chart that's drawn.

Press 2 ↓

Press ⇧ — * 8 (Hold down the shift key while you press the asterisk key)

Press C (Chart)

You'll notice that Chart automatically respaced the chart on the page, and even added a legend to distinguish one store from the other. Your new chart illustrates the month-to-month traffic at *both* stores.

Seeing your data clearly.

While this chart allows us to compare the two stores, it's still not entirely clear how the City Shop is affecting your business. What we need is some way to redesign the chart to explain this relationship. Does Chart give you that flexibility? Absolutely. Chart lets you enhance the clarity by changing any of the five components of a chart: the axes, data, labels, legend, or the chart as a whole.

Looking at our chart, it appears that the average retail traffic level was about 5000 customers per month for both you and your competitor. One way to compare the traffic in the two stores is to see how it varies from the 5000 average. Moving the horizontal axis up to the 5000 mark will enable us to see this relationship.

First you use the direction keys to tell Chart which component of the chart you want to change, then you change it with the commands listed at the bottom of your screen. Ready to change the axis?

Press [2 ↓] once

Notice that an arrow appears on your screen. It's now pointing to the horizontal axis. The arrow is one way of telling Chart what you want to change. As you press the down direction key, the arrow moves to different components of the chart.

To select the other axis, we'll need to use the right direction key. (The down direction key moves the arrow to different components of the chart; the right direction key moves the arrow to different parts of the component.)

Press [6 →] once

You should now see the arrow pointing to the vertical axis. Note the words "Axis Value" in the lower-left corner of your screen. It's another way of seeing what you're selecting to change.

Now that Chart knows exactly what part of the chart we want to change, let's look at the range of values of the vertical, or value, axis that you can change. We'll need to use the Range command.

Press [R] (Range)

Just as with other commands, the Range command shows you all of your options. For the value axis, you can change the minimum value, the maximum value, where the axes cross, or any of the options listed. Our goal is to change where the axes cross.

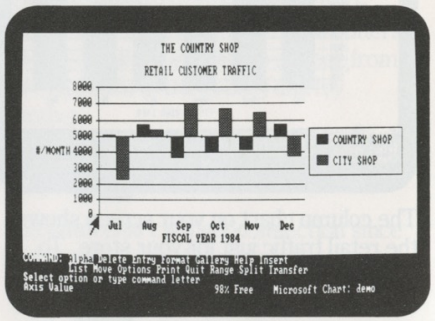
Press [Tab ⇤⇥] 8 times

You should now see the highlight marker, or cursor, next to "axis crosses" near the bottom of your screen. (If you lose count, just keep pressing the tab key until you're next to "axis crosses.") It says "0," since the axes now cross at the 0 mark. To change this crossing point:

Type **5000**

This tells Chart that we want the axes to cross at 5000.

Press [↵]



Now the relationship between the two stores is much more obvious. You can easily tell how much better than average a good month is by how far above the 5000 level it extends. When the Country Shop has good months, the City Shop has bad ones.

Because Chart lets you *see* your data, you'll be able to make much better business decisions on sales and promotions for the Country Shop. And in this case, you'll want to recommend increasing your fall promotions!

Sizing and moving things around.

Microsoft Chart places few limits on what you can change. You can change size. Color. Pattern. Position. Frame. The list goes on and on. Just by adding details to your chart, you can maintain your audience's attention. You can create professional, high-quality charts that communicate your data the way you want it communicated.

Let's look at one example of how Chart lets you move and size things to suit your needs.

We can give our chart more room if we make the legend smaller and move it to the area below the columns and above the November and December labels. But first let's tell Chart not to redraw our chart after each step. It's a lot like turning recalculation off in a spreadsheet. We'll be able to work more quickly as we revise the chart.

Press [O] (Options)

Press [N] (No)

Press [↵]

Now let's select the legend. Of course we could move and resize any part of the chart—the axes, the labels, even the chart as a whole, but in this case, moving and decreasing the size of the legend will give the chart more room to spread out.

Press [2 ↓] twice

You should now see "legend" indicated at the bottom-left corner of your screen. And, the arrow should be pointing to the legend.

The direction keys will help us out again. As you use the direction keys, watch what's happening on your screen. The legend moves in varying degrees depending on which direction keys you use.

Notice too that Chart lets you see what you're doing *while* you're doing it. You get instant feedback, so you make the right decisions the first time.

Press [M] (Move)

Press [3 Pg Dn] once

Press [2 ↓] 8 times

Press [Ctrl] — [4 ←] 3 times
(Hold the control key down while you press the left direction key)

Note: If you accidentally hit any other key besides a direction key, you can recover by doing the following:

Press Esc

Press M (Move)

Continue with the above directions, following the Move command.

Press ↵

Remember that we turned off "redraw" so Chart won't redraw the chart until told to do so. It looks as though the legend will be a little obtrusive in its new location. Let's reduce its size and *then* we'll have Chart redraw it.

Sizing chart elements is much like moving them. Chart lets you see exactly what's happening. There's no guesswork that sends you back to make more changes later on.

Press F (Format)

Press S (Size)

Again, remember to watch your screen as you use the direction keys.

Press 4 ← 10 times

Press ⇤⇥

Press 2 ↓ 7 times

Press ↵

Telling Chart to redraw your chart is simple. There's a special function key for that purpose.

Press F4

Now your chart is nicely respaced across your screen. Whenever you make a change, Chart adjusts to your changes, saving you the time and effort. You can see how easy it is to resize or move something from one place to another.

No matter what type of editing you do, with Microsoft Chart you'll edit faster and construct new presentations more easily.

The final touch.

Let's get dramatic and add a touch of class to our chart. Before we do, let's turn redraw back on.

Press O (Options)

Press Y (Yes)

Press ↵

Now for our final touch. A background shadow behind the chart will add a nice effect.

First we select the whole chart using the direction keys as we did before.

Press 2 ↓ twice

Notice that we lost the arrow and "Chart" now appears in the lower-left corner. That means that any action we take will affect the appearance of the entire chart, not just one component.

Press F (Format)

Press T (Type)

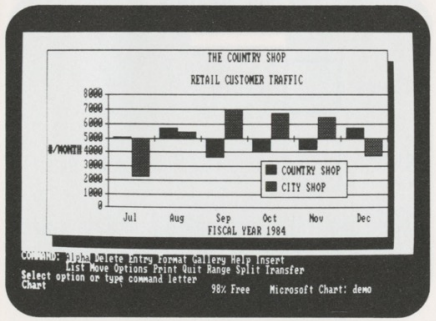
To add a shadow:

Press ⇤⇥ 10 times (to "frame type")

To select a shadow frame:

Press S (Shadow)

Press ↵



Take a minute to admire your work. As you can see, the chart has a nice shadow frame around it. In just a few minutes you've created a very persuasive presentation of the Country Shop's competitive situation.

When you customize a chart as we just did, you'll never have to repeat the same process. You can use this same chart again—with different data. You'll save time with formats that you use a lot and maintain consistency in your presentations.

Microsoft Chart also lets you overlay charts for easier comparison of your information. Switch fonts or patterns. Put in as much text as you want, anywhere you want. Draw in some arrows between your message and the chart to make your point stronger. Or print up to 16 separate charts on one page. Chart's flexibility is virtually endless!

Before we move on, let's clear our screen.

Press T (Transfer)

Press C (Clear)

Press A (All)

Press Y (Yes)

Building your own chart.

Let's say that you'd like to illustrate the important role advertising plays in your local situation when you make your presentation to the Country Shop's corporate office. The custom chart you developed makes a strong case for the competition between you and the City Shop. But to sell corporate on your fiscal year 1985 budget proposal, you need to look more specifically at the Country Shop: your advertising expenditures to sales volume.

Until now we've used data that's already been entered for you. Now it's time for you to see how easy it is to enter data on your own.

Chart lets you choose the most efficient means for entering data. You can enter data straight from the keyboard, or from virtually any popular business program, including the Multiplan electronic worksheet, Lotus 1-2-3, VisiCalc, dBASE II, or just about any other business program you like.

Let's first load your sales figures from the keyboard. In a minute we'll show you more about entering data from other programs, using Multiplan as an example.

Creating a data series from the keyboard.

Press [L] (List)

What you see on your screen may not make a lot of sense to you right now. Don't worry, it will in a minute.

There are two steps to creating your own data series. First you name it, and then you type in the data.

Press [N] (Name)

There are several things to indicate when you're naming a data series. The first is the series name.

Type **Sales** (Do *not* press the return key)

If you make a mistake when you're typing, you can just press the backspace key and then retype the name.

Next tell Chart how your data will be categorized, or labeled, on the "x" axis. Let's call it Fiscal Year 1984.

Press [⇤⇥] once (to "category name")

Right now it says "category." Let's replace that with something a little more descriptive.

Type **FY 1984**

Press [⇤⇥] once (to "value name")

The highlight marker is now after "value name." This is the label for the "y" axis. Since we're charting our sales figures, we'll indicate that the numbers represent dollars.

Type **Dollars**

Press [⇤⇥] once (to "category type")

Let's tell Chart that we're working with dates. You'll see the time-saving advantage of this in a minute.

Press [D] (Date)

Press [⇤⇥] once (to "category start")

Next we tell Chart when our data starts. Go ahead and type over what's there.

Type **7/1/83**

Press [⇤⇥] once (to "increment")

As you've just seen, there are times when Chart will propose a response. Because each of your categories differs by one month, there is no need to replace the proposed response of "1" in this case.

Press [⇤⇥] once (to "increment period")

Press [M] (Month)

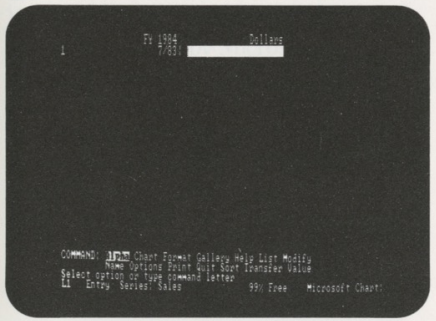
Press [↵]

The name of your series is displayed and its type identified as Date. The "Source of Data" column is blank and the "# pts.," or number of points, column shows "0" because you have not entered your data yet.

Entering the data is the next step you'll take in creating your own data series. To do so, you'll need to move to the entry screen.

Entering data.

Press [E] (Entry)



Now the entry screen has replaced the list screen. Because Chart already knows the categories, you only need to type the monthly sales volume. The value column on the first line is highlighted, awaiting your first entry.

The Country Shop collected $532,173 in July, 1983.

To enter that amount:

Type **532173**

Press [2↓]

Chart is ready for your next entry. Now you can see the advantage of indicating "dates." Instead of typing each date separately, Chart knows the dates we'll need.

Type **621818**

Press [2↓]

Enter the following sales figures and press the down direction key after each entry:

Type **540012**

Press [2↓]

Type **533231**

Press [2↓]

Type **546547**

Press [2↓]

To enter your final value:

Type **684020** (Do not press the down direction key)

> Note: If you accidentally pressed the down direction key, you can easily recover from this mistake.
>
> Press [8↑]
>
> Continue with the demonstration.

Press [↵]

Suppose you've changed your mind and want to change the format of the dates you've already entered. Does that mean

we have to reenter all the data? Of course not. Chart lets you change your mind as often as you'd like without starting over. Let's change the way the dates are written.

Press F (Format)

Press C (Categories)

We'll need to tab down to the "date format."

Press ⇤⇥ 8 times (to "date format")

Chart wants to know if you'd like the short, medium, or long version for representing dates. Forget which is which? Easy. Just ask Chart for help.

Help is at your fingertips.

Press Alt — H (Hold down the Alt key while you press the H key)

You are now looking at specific information about date formats. Chart's on-line Help always brings you right to the information you need. No hunting around, or endless paging through screen after screen of information. As you can see, Help clearly defines the differences between short, medium, and long dates. (Let's choose the medium style when we continue.)

You can always gather additional information once you're in Help. Notice that there is a special menu of commands at the bottom of your screen. You use these commands to move around in the Help file and to return to your chart once you have the answers to your questions. For instance, you can move to the next page.

Press N (Next)

Or, to the keyboard guide.

Press K (Keyboard)

And, you can always go back to where you were with the Resume command.

Press R (Resume)

Now you're right back where you left off, ready to continue with the information you need about short, medium, and long date formats.

Press M (Medium)

Remember that we labeled our category, "FY 1984," so it's probably not necessary to show the year at every point along the axis.

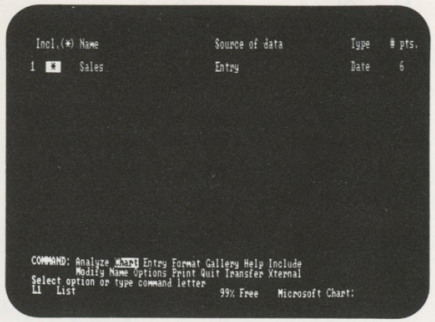
Press ⇤⇥ (to "show year")

Press N (No)

Once you return to your "entry" screen, you'll notice that Chart automatically changed all of the dates for you. Where it now says 7/83, it will say "Jul." Just one more example of how Microsoft Chart lets you work quickly.

Press ↵

Press L (List)



That's all there is to it. You can now see that the list screen shows you what lists of data you have. "Sales" is listed as a complete data series. And, under "Source of Data," Chart tells you that you've used the "Entry" screen to enter your data.

You are now familiar with the three screens displayed by Microsoft Chart. The entry screen is where you enter data, the list screen is where you manipulate and analyze whole data series, and the chart screen is where you see your chart. You enter and edit data, and make and revise charts, by moving freely among these three screen displays.

Linking up with Multiplan.

Now that we're all set with our sales figures, we need to enter the Country Shop's advertising expenditures to see how the two compare. Let's suppose you've already recorded these costs on a Multiplan electronic worksheet. If you are not an owner of the Microsoft Multiplan electronic worksheet but would like more information about it, contact your local computer dealer or call 1-800-426-9400; in Washington State, Alaska, Hawaii, or Canada call 206-828-8088.

The first step is to name your data series, just like we did with the sales data. And then you have to load in the data from your Multiplan file.

Press 2↓ once

As you can see, Chart is ready for your second data series.

Press N (Name)

Let's keep it simple and name our second series "Ad Dollars."

Type **Ad Dollars**

We'll move through these next steps rather quickly, since we just went through similar steps. Next we need to tell Chart how our data will be categorized.

Press ⇤⇥ once (to "category name")

Type **FY 1984**

Press ⇤⇥ once (to "value name")

Again, we're charting costs.

Type **Dollars**

Press ⇤⇥ once (to "category type")

And, we're working with dates.

Press D (Date)

Press ⇤⇥ (to "category start")

Type **July 1, 1983**

Press [⇤⇥] twice (to "increment period")

Press [M] (Month)

Press [↵]

When you copy data from Multiplan, it's just like linking worksheets within Multiplan itself. So if you're already familiar with the Multiplan electronic worksheet, you're one step ahead.

Now tell Chart that the data is from an external source.

Press [X] (Xternal)

By pressing the right direction key, you'll see a list of file names to choose from. This is especially helpful when you don't remember the name of your file.

Press [6 →] once

The file we're after is called CSHOPADS for Country Shop Advertising.

Selecting the file you want is simple. Just—

Press [6 →] once until "CSHOPADS" is highlighted (Do not hit the return key)

CSHOPADS is the name of your Multiplan worksheet. Since most worksheets contain quite a bit of data, it's important to name portions of the worksheet that you'll want to use. We've already labeled the rows and columns where your ad costs were recorded.

Press [⇤⇥]

Now Chart automatically shows you all the portions of your worksheet that you've named. Since there's only one entry, all you have to do is press the return key.

Press [↵]

You can even establish a permanent link between your Multiplan worksheet and Chart. If you change numbers in Multiplan, the next time you look at your chart it will automatically be updated. You'll never need to spend time reentering data manually!

Chart works with any software package that supports DIF, ASCII, or SYLK file formats. Which means you can chart data contained in most spreadsheets, databases, financial accounting packages, or programs like Microsoft Project.

Looking at your results.

What type of chart is just right for this data? We could select one of the 45 possibilities from the gallery, then custom tailor it to our needs as we have done. Yet, there's another possibility. Chart has a special feature that lets you reuse formats again and again. It's a great way to create high-quality business graphics without having to duplicate steps. We designed a format that we think you'll find useful. Let's give it a try.

Press [T] (Transfer)

Press [L] (Load)

Type **special** (Do *not* press the return key)

Before we enter the command by pressing the return key, we need to tell Chart to load only the format, since we already have our data.

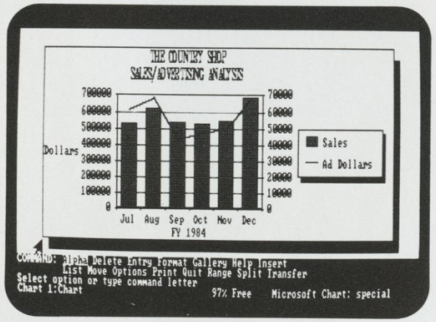
Press [⇤⇥]

Press [F] (Format)

Press [↵]

Ready to see your chart?

Press [C] (Chart)



You now have a very impressive chart. This "mixed" chart shows a definite correlation between your advertising expenditures and sales volume—a point that you can leverage to increase your advertising budget in fiscal year 1985.

Because Microsoft Chart is compatible with a wide range of printers and plotters, you can select the one that best suits your needs. If you'd like to print or plot the chart on your screen for your presentation to corporate, there are specific instructions in the Chart manual and the on-line Help file on how to do so.

A few more samples.

Before you complete this demonstration, take a few minutes to review some additional charts we designed with Microsoft Chart. They'll show you the some of the more advanced capabilities you'll enjoy.

First—

Press [T] (Transfer)

Press [L] (Load)

Type **Orders84**

Press [↵]

To see the other charts, follow the steps above, typing in any one of the following names instead of "Orders84": "Chemical," "MFCorp," or "PetroCh."

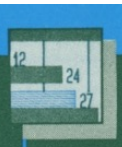
This concludes your guided tour of Microsoft Chart. We hope you agree that Microsoft Chart is the business graphics program that gives you the flexibility to sell your ideas precisely the way you need to. It's an excellent way to get your point across faster and more clearly.

We encourage you to review any of the parts of the demonstration in whatever order you like. Plus, you may find that you'd like to experiment with different chart formats in the gallery. Do so freely, using the files listed above or the files named "demo" and "part2."

When you're finished, follow these steps to quit the program.

Press [Q] (Quit)

Press [Y] (Yes)



MICROSOFT®

MICROSOFT CORPORATION
10700 NORTHUP WAY
BOX 97200
BELLEVUE, WA 98009
USA